

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 06-095874

(43)Date of publication of application : 08.04.1994

(51)Int.Cl. G06F 9/30
G06F 9/34

(21)Application number : 05-165052

(71)Applicant : INTERNATL BUSINESS MACH
CORP <IBM>

(22)Date of filing : 11.06.1993

(72)Inventor : JAMES EDWARD PHILIP
VASSILIADIS STAMATIS

(30)Priority

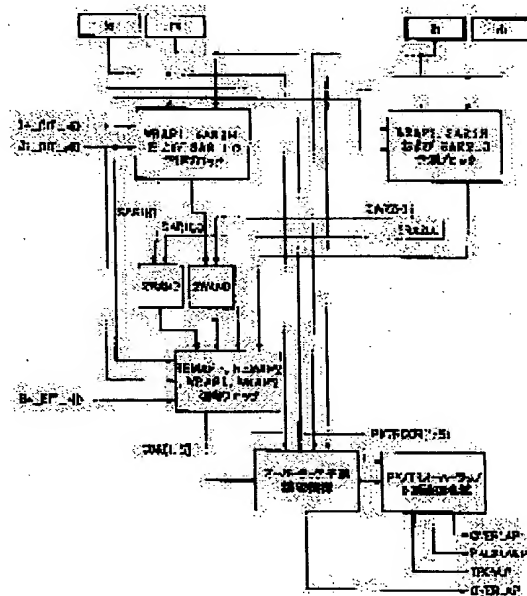
Priority number : 92 920941 Priority date : 28.07.1992 Priority country : US

(54) DIGITAL COMPUTER SYSTEM

(57)Abstract:

PURPOSE: To provide a device which detects the stored operand overlap for SS instruction which has the same overlap detection condition as a character movement (MVC) instruction and to prove that this device is effective for this detection.

CONSTITUTION: This device is applied to all ESA/390 addressing modes including an access register addressing mode for 24-bit or 31-bit addressing. 24-Bit or 31-bit S/370 addressing is handled as a special case of access register addressing. Further, this device is extended so as to support other addressing modes including the 64-bit addressing mode. This device is provided with a high-speed parallel execution mode also.



LEGAL STATUS

[Date of request for examination] 11.06.1993

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 2500098

[Date of registration] 01.03.1996

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平6-95874

(43)公開日 平成6年(1994)4月8日

(51)Int.Cl. ⁵	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 9/30	3 5 0 B	9189-5B		
9/34	3 2 0 A	9189-5B		

審査請求 有 請求項の数12(全 42 頁)

(21)出願番号 特願平5-165052

(22)出願日 平成5年(1993)6月11日

(31)優先権主張番号 9 2 0 9 4 1

(32)優先日 1992年7月28日

(33)優先権主張国 米国 (U S)

(71)出願人 390009531

インターナショナル・ビジネス・マシーンズ・コーポレーション

INTERNATIONAL BUSINESS MACHINES CORPORATION

アメリカ合衆国10504、ニューヨーク州
アーモンク (番地なし)(72)発明者 ジェームズ・エドワード・フィリップス
アメリカ合衆国13905、ニューヨーク州
ペンガムトン、クレイリー・アベニュー138

(74)代理人 弁理士 合田 潔 (外1名)

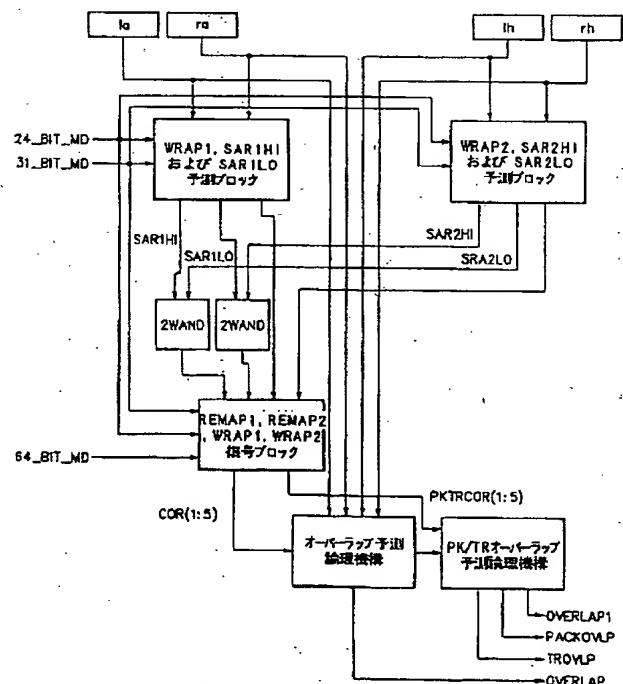
最終頁に続く

(54)【発明の名称】 デジタル・コンピュータ・システム

(57)【要約】 (修正有)

【目的】 文学移動(MVC)命令と同じオーバーラップ検出条件を有するSS命令用の記憶オペランド・オーバーラップを検出するための装置を提示し、それに有効なことを証明すること。

【構成】 この装置は、24ビットまたは31ビット・アドレス指定用のアクセス・レジスタ・アドレス指定を含むすべてのESA/390アドレス指定モードに適用できる。24ビット及び31ビットのS/370アドレス指定もアクセス・レジスタ・アドレス装置の特殊なケースとして扱われる。さらに、この装置は、他のアドレス装置モードをサポートするように拡張され、64ビット・アドレス指定モードを含む例が示される。この装置の高速並列実施態様も提示される。



【 特許請求の範囲】

【 請求項1 】 記憶オペランド・オーバーラップ予測機構と、

宛先記憶オペランド・アドレス低及び折返し検出器と、
ソース記憶オペランド・アドレス高及び折返し検出器と、

オーバーラップ検出器用の制御論理機構とを備えるデジタル・コンピュータ・システム。

【 請求項2 】 オーバーラップが、宛先記憶オペランド・アドレスとソース記憶オペランド・アドレスの生成に使用される入力ならびにオペランド長さから予測され、ソース記憶オペランド・アドレス及び宛先記憶オペランド・アドレスの計算と並列に実行されることを特徴とする、請求項1 に記載のデジタル・コンピュータ・システム。

【 請求項3 】 前記宛先記憶オペランド・アドレス低及び折返し検出器中で、宛先オペランド・アドレスが仮想ア *

$$SAR1H_i = Ia_i \forall ra_{(i-52)} \quad 52 \leq i \leq 55$$

$$SAR1G_{i_N} = \overline{Ia_i ra_{(i-32)}} \quad 52 \leq i \leq 62$$

$$SAR1T_{i_N} = \overline{Ia_i + ra_{(i-52)}} \quad 52 \leq i \leq 62$$

$$SAR1C_{63_N1} = \overline{Ia_{63} ra_{11}}$$

$$SAR1H_{i_N} = \overline{SAR1H_i} \quad 52 \leq i \leq 55$$

$$(SAR1G)_{i+1} = \overline{(SAR1G)_{i_N} ((SAR1T)_{i_N} + (SAR1G)_{i+1_N})} \quad 52 \leq i \leq 61$$

$$(SAR1T)_{i+1} = \overline{((SAR1T)_{i_N} + (SAR1T)_{i+1_N})} \quad 52 \leq i \leq 61$$

$$(SAR1C1)_{62} = \overline{(SAR1G)_{62_N} ((SAR1T)_{62_N} + (SAR1C)_{63_N1})}$$

$$(SAR1C1)_{63} = \overline{(SAR1C)_{63_N1}}$$

$$(SAR1G)_{i+2_N} = \overline{(SAR1G)_{i+1} + (SAR1T)_{i+1} (SAR1G)_{i+2}} \quad 52 \leq i \leq 59$$

$$(SAR1T)_{i+2_N} = \overline{(SAR1T)_{i+1} (SAR1T)_{i+2}} \quad 52 \leq i \leq 59$$

$$(SAR1C)_{i_N2} = \overline{(SAR1G)_{i+1} + (SAR1T)_{i+1} (SAR1C1)_{i+2}} \quad i = 60, 61$$

$$(SAR1C)_{i_N2} = \overline{(SAR1C1)_i} \quad i = 62, 63$$

$$(SAR1G)_{i+7} = \overline{(SAR1G)_{i+2_N} ((SAR1T)_{i+2_N} + (SAR1G)_{i+7_N})} \quad 52 \leq i \leq 55$$

$$(SAR1T)_{i+7} = \overline{(SAR1T)_{i+2_N} + (SAR1T)_{i+7_N}} \quad 52 \leq i \leq 55$$

$$(SAR1C)_{55} = \overline{(SAR1G)_{55_N} ((SAR1T)_{55_N} + (SAR1C)_{60_N2})} \quad 56$$

$$(SAR1C)_i = \overline{(SAR1C)_{i_N2}} \quad 60 \leq i \leq 63$$

$$(SAR1C)_{i_N} = \overline{(SAR1G)_{i+7} + (SAR1T)_{i+7} (SAR1C)_{i+7}} \quad 52 \leq i \leq 55$$

$$(SAR1C)_{55_N} = \overline{(SAR1C)_{55}}$$

$$SAR1_{i_N} = (SAR1C)_{i+1_N} \vee SAR1H_i \quad 52 \leq i \leq 55$$

$$SAR1_i = (SAR1C)_{i+1_N} \vee SAR1H_{i_N} \quad 52 \leq i \leq 55$$

$$SAR1_{52_to_55_ZEROS} = (SAR1_{52_N}) (SAR1_{53_N}) (SAR1_{54_N}) (SAR1_{55_N})$$

$$SAR1_{52_to_55_ONES} = (SAR1_{52}) (SAR1_{53}) (SAR1_{54}) (SAR1_{55})$$

$$(SAR1C)_{52} = \overline{(SAR1C)_{52_N}}$$

上式において、SAR1H_iは、ビット位置iにおける半和を表し、SAR1G_i_Nは生成なしを表し、SAR1T_i_Nは伝送なしを表し、(SAR1C)_i_N

*ドレスの低位256バイト・ブロック中にあること、及び仮想アドレス空間の高位4Kブロックから仮想アドレス空間の低位4Kブロックへの折返し動作がアドレス生成中に起こるかどうか判定されることを特徴とする、請求項1 に記載のデジタル・コンピュータ・システム。

【 請求項4 】 前記宛先記憶オペランド・アドレス低及び折返し検出器中で、64ビット入力I_aに宛先記憶オペランド・アドレスの算出に使用される基底が供給され、12ビット入力r_aに同じく変位が供給されることを特徴とする、請求項1 に記載のデジタル・コンピュータ・システム。

【 請求項5 】 前記宛先記憶オペランド・アドレス低及び折返し検出器が12ビット加算器を含み、その中での挙動が下記の数式1 のブール式によって決定され、

【 数1 】

1 は緩衝キャリーなしを表し、(SAR1C1)_iは緩衝キャリーを表し、(SAR1C)_iはキャリーを表し、(SAR1C)_i_Nはキャリーなしを表し、SA

3

R1HI_Nは半和なしを表し、

【数2】

(SAR1G)_N

は、ビット位置_nから_mにわたるグループに関する、グループ生成を表し、

【数3】

(SAR1T)_N

はグループ伝送を表し、

【数4】

(SAR1G)_N

はグループ生成なしを表し、

【数5】

(SAR1T)_N

はグループ伝送なしを表し、SAR1_iはビット位置_i *
 $la_iN = \overline{la_i}$

$$la_iN = \overline{la_i}$$

$$la_{40_TO_51_ZEROS} = la_{40_N} la_{41_N} \dots la_{50_N} la_{51_N}$$

$$la_{40_TO_50_ONES} = la_{40} la_{41} \dots la_{50}$$

$$la_{40_TO_51_ONES} = la_{40_TO_50_ONES} la_{51}$$

$$la_{40_TO_50_ONES_A_51_ZERO} = la_{40_TO_50_ONES} la_{51_N}$$

$$la_{33_TO_39_ZEROS} = la_{33_N} la_{34_N} \dots la_{39_N} la_{39_N}$$

$$la_{33_TO_39_ONES} = la_{33} la_{34} \dots la_{39} la_{39}$$

$$la_{33_TO_50_ONES_A_51_ZERO} = la_{33_TO_39_ONES} la_{40_TO_50_ONES_A_51_ZERO}$$

$$la_{33_TO_51_ZEROS} = la_{33_TO_39_ZEROS} la_{40_TO_51_ZEROS}$$

$$la_{33_TO_51_ONES} = la_{33_TO_39_ONES} la_{40_TO_51_ONES}$$

$$SAR1_TO_51_0S = 24_BIT_MD la_{40_TO_51_ZEROS} + 31_BIT_MD la_{33_TO_51_ZEROS}$$

$$SAR1_TO_51_1S = 24_BIT_MD la_{40_TO_51_ONES} + 31_BIT_MD la_{33_TO_51_ONES}$$

$$SAR1_TO_50_1S_A_51_0 = 24_BIT_MD la_{40_TO_50_ONES_A_51_ZERO} +$$

$$31_BIT_MD la_{33_TO_50_ONES_A_51_ZERO}$$

$$SAR1LO = [(SAR1C)_{32_N} SAR1_TO_51_0S + (SAR1C)_{32} SAR1_TO_51_1S] SAR1_52_TO_55_ZEROS$$

$$SAR1HI = [(SAR1C)_{32_N} SAR1_TO_51_1S + (SAR1C)_{32} SAR1_TO_50_1S_A_51_0] SAR1_52_TO_55_ONES$$

$$WRAP1 = (SAR1C)_{32} SAR1_TO_51_1S$$

$$WRAP1_N = \overline{WRAP1}$$

上式において、 la_iN はビット位置_iにおける入力
 の1の補数を表し、 $la_{40_TO_51_ZERO}$
 Sは、40と51の間の両端を含むすべてのビット位置
 で入力_aがすべて0であることを表し、 $la_{40_TO_50_ONES}$ は、40と50の間の両端を含む
 40
 すべてのビット位置で入力_aがすべて1であることを
 表し、 $la_{40_TO_51_ONES}$ は、40と51
 の間の両端を含むすべてのビット位置で入力_aがす
 べて1であることを表し、 $la_{40_TO_50_ONES_A_51_ZERO}$ は、40と50の間の両端
 を含むすべてのビット位置で入力_aがすべて1.であ
 り、ビット位置51では_aが0であることを表し、 $la_{33_TO_39_ZEROS}$ は、33と39の間
 の両端を含むすべてのビット位置で入力_aがすべて0 50

4

*における加算の結果を表し、さらにSAR1_{52_to_55_ZEROS}はビット位置52から55までの
 すべての結果ビットがゼロであることを表し、SAR1_{52_to_55_ONES}はビット位置52から5
 5までのすべての結果ビットが1であることを表すこと
 を特徴とする、請求項1に記載のデジタル・コンピュ
 ータ・システム。

【請求項6】前記宛先記憶オペランド・アドレス低及び
 折返し検出器が、ビットごとのインバータを含み、アド
 レス指定モードの指標24_BIT_MDまたは31_BIT_MD
 10
 の制御下で多元AND及び多重化を行っ
 て、下記の数式6のブール式を計算することによりSA
 R1LO及びWRAP1を生成し、

【数6】

$$33 \leq i \leq 51$$

であることを表し、 $la_{33_TO_39_ONES}$
 は、33と39の間の両端を含むすべてのビット位置で
 入力_aがすべて1であることを表し、 $la_{33_TO_50_ONES_A_51_ZERO}$ は、33と5
 0の間の両端を含むすべてのビット位置で入力_aがす
 べて1であり、ビット位置51では_aが0であることを
 表し、 $la_{33_TO_51_ZEROS}$ は、33
 と51の間の両端を含むすべてのビット位置で入力_a
 がすべて0であることを表し、 $la_{33_TO_51_ONES}$ は、33と51の間の両端を含むすべてのビ
 ット位置で入力_aがすべて1であることを表し、24_BIT_MDは、24ビット・アドレス指定モードが
 有効であることを表し、31_BIT_MDは31ビッ
 ト・アドレス指定モードが有効であることを表し、SA

5

R1 __TO__51 __OS は、24ビット・アドレス指定の場合は40と51、31ビット・アドレス指定の場合は33と51の間の両端を含むすべてのビット位置で入力aがすべて0であることを表し、SAR1 __TO__51 __1S は、24ビット・アドレス指定の場合は40と51、31ビット・アドレス指定の場合は33と51の間の両端を含むすべてのビット位置で入力aがすべて1であることを表し、SAR1 __TO__50 __1S __A__51 __0 は、24ビット・アドレス指定では40と50の間の両端を含むすべてのビット位置で入力aがすべて1でビット位置51では1であり、31ビット・アドレス指定では33と50の間の両端を含むすべてのビット位置で入力aがすべて1でビット位置51では1であることを表し、SAR1LOは、SAR1が仮想アドレス空間の低位256バイト・ブロック中にあることを表し、WRAP1は、ソース・アドレス計算がアドレス計算中に高位仮想メモリから低位仮想メモリへ折り返すことを表すことを特徴とする、請求項1に記載のデジタル・コンピュータ・システム。

*

$$\begin{aligned}
 SAR2H_i &= lh_i \vee rh_{(i-52)} & 52 \leq i \leq 55 \\
 SAR2G_i_N &= \overline{lh_i} rh_{(i-52)} & 52 \leq i \leq 62 \\
 SAR2T_i_N &= \overline{lh_i} + rh_{(i-52)} & 52 \leq i \leq 62 \\
 SAR2C_{63_N1} &= \overline{lh_{63}} rh_{11} \\
 SAR2H_i_N &= \overline{SAR2H_i} & 52 \leq i \leq 55 \\
 (SAR2G)_i^{+1} &= \overline{(SAR2G)_i_N ((SAR2T)_i_N + (SAR2G)_{i+1_N})} & 52 \leq i \leq 61 \\
 (SAR2T)_i^{+1} &= \overline{((SAR2T)_i_N + (SAR2T)_{i+1_N})} & 52 \leq i \leq 61 \\
 (SAR2C1)_{62} &= \overline{(SAR2G)_{62_N} ((SAR2T)_{62_N} + (SAR2C)_{63_N1})} \\
 (SAR2C1)_{63} &= \overline{(SAR2C)_{63_N1}} \\
 (SAR2G)_i^{+3_N} &= \overline{(SAR2G)_i^{+1} + (SAR2T)_i^{-1} (SAR2G)_{i+2}} & 52 \leq i \leq 59 \\
 (SAR2T)_i^{+3_N} &= \overline{(SAR2T)_i^{+1} (SAR2T)_{i+2}} & 52 \leq i \leq 59 \\
 (SAR2C)_i_N2 &= \overline{(SAR2G)_i^{+1} + (SAR2T)_i^{+1} (SAR2C1)_{i+2}} & i = 60, 61 \\
 (SAR2C)_i_N2 &= \overline{(SAR2C1)_i} & i = 62, 63 \\
 (SAR2G)_i^{-1} &= \overline{(SAR2G)_i^{-3_N} ((SAR2T)_i^{+3_N} + (SAR2G)_{i+2_N})} & 52 \leq i \leq 55 \\
 (SAR2T)_i^{-1} &= \overline{(SAR2T)_i^{-3_N} + (SAR2T)_{i+2_N}} & 52 \leq i \leq 55 \\
 (SAR2C)_{56} &= \overline{(SAR2G)_{56_N} ((SAR2T)_{56_N} + (SAR2C)_{60_N2})} & 56 \\
 (SAR2C)_i &= \overline{(SAR2C)_i_N2} & 60 \leq i \leq 63 \\
 (SAR2C)_i_N &= \overline{(SAR2G)_i^{-1} + (SAR2T)_i^{-1} (SAR2C)_{i+4}} & 52 \leq i \leq 55 \\
 (SAR2C)_{56_N} &= \overline{(SAR2C)_{56}} \\
 SAR2_i_N &= (SAR2C)_{i+1_N} \vee SAR2H_i & 52 \leq i \leq 55 \\
 SAR2_i &= (SAR2C)_{i+1_N} \vee SAR2H_i_N & 52 \leq i \leq 55 \\
 SAR2_52_to_55_ZEROS &= (SAR2_{52_N}) (SAR2_{53_N}) (SAR2_{54_N}) (SAR2_{55_N}) \\
 SAR2_52_to_55_ONES &= (SAR2_{52}) (SAR2_{53}) (SAR2_{54}) (SAR2_{55}) \\
 (SAR2C)_{52} &= \overline{(SAR2C)_{52_N}}
 \end{aligned}$$

上式において、SAR2H_iは、ビット位置iにおける 50 半和を表し、SAR2G_i__Nは生成なしを表し、SA

6

*【請求項7】前記ソース記憶オペランド・アドレス高及び折返し検出器中で、ソース・オペランド・アドレスが仮想アドレスの高位256バイト・ブロック中にあること、及び仮想アドレス空間の高位4Kブロックから仮想アドレス空間の低位4Kブロックへの折返し動作がアドレス生成中に起こるかどうか判定されることを特徴とする、請求項1に記載のデジタル・コンピュータ・システム。

【請求項8】前記ソース記憶オペランド・アドレス高及び折返し検出器中で、64ビット入力hに宛先記憶オペランド・アドレスの算出に使用される基底が供給され、12ビット入力rに同じく変位が供給されることを特徴とする、請求項1に記載のデジタル・コンピュータ・システム。

【請求項9】前記ソース記憶オペランド・アドレス高及び折返し検出器が12ビット加算器を含み、その中の挙動が下記の数式7のブール式によって決定され、

【数7】

R2Ti_i__Nは伝送なしを表し、(SAR2C)_i__N
1は緩衝キャリアなしを表し、(SAR2C1)_iは緩
衝キャリアを表し、(SAR2C)_iはキャリアを表
し、(SAR2C)_i__Nはキャリアなしを表し、SA
R2Hi__Nは半和なしを表し、

【数8】

(SAR2G) **B**

は、ビット 位置 n から m にわたるグループに関する、グループ生成を表し、

【数9】

(S A R 2 T) m

はグループ伝送を表し、

【数10】

(SAR2G) N

はグループ生成なしを表し、

【数1 1】

$$Ih_i \cdot N = \overline{Ih_i}$$

$$lh_40_TO_51_ZEROS = lh_{40_N} lh_{41_N} \dots lh_{50_N} lh_{51_N}$$

$$lh_40_TO_50_ONES = lh_{40} lh_{41} \dots lh_{50}$$

$$lh_40_TO_51_ONES = lh_40_TO_50_ONES \, lh_{51}$$

$$lh_40_TO_50_ONES_A_51_ZERO = lh_40_TO_50_ONES\ lh_{51}_N$$

$$lh_33_TO_39_ZEROS = lh_{33_N} lh_{34_N} \dots lh_{38_N} lh_{39_N}$$

$$lh\ 33_TO_39_ONES = lh_{33}\ lh_{34} \dots lh_{38}\ lh_{39}$$

$$lh_33_TO_50_ONES_A_51_ZERO = lh_33_TO_39_ONES\ lh_40_TO_50_ONES_A_51_ZERO$$

$$lh_33_TO_51_ZEROS = lh_33_TO_39_ZEROS \parallel lh_40_TO_51_ZEROS$$

lh_33_TO_51_ONES = lh_33_TO_39_ONES lh_40_TO_51_ONES

$$SAR2_TO_51_OS = 24_BIT_MD1h_40_TO_51_ZEROS + 31_BIT_MD1h_33_TO_51_ZEROS$$

$$SAR2_TO_51_1S = 24_BIT_MD1h_40_TO_51_ONES + 31_BIT_MD1h_33_TO_51_ONES$$

SAR2_TO_50_1S_A_51_0 = 24_BIT_MD1h_40_TO_50_ONES_A_51_ZERO +

31_BIT_MD1h_33_TO_50_ONES_A_51_ZERO

$$SAR2LO = [(SAR2C)_{52_N} SAR2_TO_51_0S + (SAR2C)_{52} SAR2_TO_51_1S] SAR2_52_TO_55_ZEROS$$

```
SAR2HI = [(SAR2C)s2_N SAR2_TO_51_1S + (SAR2C)s2 SAR2_TO_50_1S_A_51_0] SAR2_52_to_55_ONES
```

WRAP2 = (SAR2C)₅₂ SAR2_TO_51_1S

$$\text{WRAP2 } N = \overline{\text{WRAP2}}$$

上式において、 $1\ h_i_N$ はビット位置 i における入力
の 1 の補数を表し、 $1\ h_40_TO_51_ZERO$
S は、 40 と 51 の間の両端を含むすべてのビット位置
で入力 h がすべて 0 であることを表し、 $1\ h_40_TO_50_ONES$
は、 40 と 50 の間の両端を含む
すべてのビット位置で入力 h がすべて 1 であることを
表し、 $1\ h_40_TO_51_ONES$ は、 40 と 51
の間の両端を含むすべてのビット位置で入力 h がす
べて 1 であることを表し、 $1\ h_40_TO_50_ONES_A_51_ZERO$
は、 40 と 50 の間の両端
を含むすべてのビット位置で入力 h がすべて 1 であ
り、ビット位置 51 では a が 0 であることを表し、 1

(SARZT) N

*はグループ伝送なしを表し、SAR2_iはビット位置iにおける加算の結果を表し、さらにSAR2_52_t o _55_ZEROSはビット位置52から55までのすべての結果ビットがゼロであることを表し、SAR2_52_t o _55_ONESはビット位置52から55までのすべての結果ビットが1であることを表すことを特徴とする、請求項1に記載のデジタル・コンピュータ・システム。

10 【請求項10】前記ソース記憶オペランド・アドレス高及び折返し検出器が、ビットごとのインバータを含み、アドレス指定モードの指標24 __BI T__MDまたは31 __BI T__MDの制御下で多元AND及び多重化を行って、下記の数式12のブール式を計算することによりSAR2H1及びWRAP2を生成し、

* 【数1 2】

$$33 \leq i \leq 51$$

40 h _33 _TO_39 _ZEROS は、33 と 39 の間の両端を含むすべてのビット位置で入力 h がすべて0であることを表し、1 h _33 _TO_39 _ONES は、33 と 39 の間の両端を含むすべてのビット位置で入力 h がすべて1であることを表し、1 h _33 _TO_50 _ONES _A_51 _ZERO は、33 と 50 の間の両端を含むすべてのビット位置で入力 a がすべて1であり、ビット位置 51 では h が0であることを表し、1 h _33 _TO_51 _ZEROS は、33 と 51 の間の両端を含むすべてのビット位置で入力 h がすべて0であることを表し、1 h _33 _TO_51
50 ONES は、33 と 51 の間の両端を含むすべてのビ

9

ット位置で入力 h がすべて1であることを表し、24 _BIT_MD は、24ビット・アドレス指定モードが有効であることを表し、31 _BIT_MD は31ビット・アドレス指定モードが有効であることを表し、SAR2 _TO_51_OS は、24ビット・アドレス指定の場合は40と51、31ビット・アドレス指定の場合は33と51の間の両端を含むすべてのビット位置で入力 h がすべて0であることを表し、SAR2 _TO_51_1S は、24ビット・アドレス指定の場合は40と51、31ビット・アドレス指定の場合は33と51の間の両端を含むすべてのビット位置で入力 h がすべて1であることを表し、SAR2 _TO_50_1S_A_51_0 は、24ビット・アドレス指定では40と *

$$\text{REMAP1_N} = \text{SAR2HI SAR1LO}$$

$$\text{REMAP1} = \overline{\text{REMAP1_N}}$$

$$\text{COR}(1) = 64_BIT_MD + \text{REMAP1_N WRAP1_N WRAP2_N} + \text{WRAP2 WRAP1}$$

$$\text{COR}(2) = 24_BIT_MD \text{ REMAP1_N WRAP2_N WRAP1}$$

$$\text{COR}(3) = 24_BIT_MD \text{ WRAP2 WRAP1_N} + 24_BIT_MD \text{ REMAP1 WRAP1_N}$$

$$\text{COR}(4) = 31_BIT_MD \text{ REMAP1_N WRAP2_N WRAP1}$$

$$\text{COR}(5) = 31_BIT_MD \text{ WRAP2 WRAP1_N} + 31_BIT_MD \text{ REMAP1 WRAP1_N}$$

上式において、24 _BIT_MD は、24ビット・アドレス指定が有効なことを表し、31 _BIT_MD は、31ビット・アドレス指定が有効なことを表し、REMAP1とREMAP1 _N は、宛先記憶アドレスが仮想アドレス空間の低位256バイト・ブロックにあり、ソース記憶アドレスが仮想アドレス空間の高位256バイト・ブロックにある真と補を表し、COR(1)はMVC型オーバーラップを予測するための可能な5つの計算のうちの第1の計算を表し、COR(2)はMVC型オーバーラップを予測するための可能な5つの計算のうちの第2の計算を表し、COR(3)はMVC型オーバーラップを予測するための可能な5つの計算のうちの第3の計算を表し、COR(4)はMVC型オーバーラップを予測するための可能な5つの計算のうちの第4の計算を表し、COR(5)はMVC型オーバーラップを予測するための可能な5つの計算のうちの第5の計算を表すことを特徴とする、請求項1に記載のデジタル・コンピュータ・システム。

【請求項12】前記記憶オペランド・オーバーラップ予測機構が、

入力 a 、 $l h$ 、 $r a$ 、 $r h$ 、COR(1:5)を備える予測論理機構と、

複数のキャリーセーブ加算器CSA1、CSA2、CSA3、CSA4、CSA5、CSA6、CSA7と、

キャリー先読み加算器CLA1と、

キャリー生成器CG1と、

複数の排他的論理和ブロックXR7、XR8と、

複数の多元ANDブロックA1、A2、A3、A4、A

10

*50の間の両端を含むすべてのビット位置で入力 h がすべて1でありビット位置51では1であり、31ビット・アドレス指定では33と50の間の両端を含むすべてのビット位置で入力 h がすべて1でありビット位置51では1であることを表し、SAR2HIは、SAR2が仮想アドレス空間の低位256バイト・ブロック中にあることを表し、WRAP2は、ソース・アドレス計算がアドレス計算中に高位仮想メモリから低位仮想メモリへ折り返すことを表すことを特徴とする、請求項1に記載のデジタル・コンピュータ・システム。

【請求項11】前記オーバーラップ検出器用の制御論理の挙動が下記の数式13のブール式によって決定され、

【数13】

5と、

OVERLAPを計算するための多重化論理機構とを有することを特徴とする、請求項1に記載のデジタル・コンピュータ・システム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、デジタル・コンピュータ・システムに関し、詳細にはソース位置と宛先位置の破壊的オーバーラップを予測し、それによって条件の決定が計算できるようになるまでパイプライン式デジタル・コンピュータ・システムにおいて実行を停止する必要なく、即時に移動が実行できるようにすることにより、ある記憶位置から第2の記憶位置への頻繁に使用される移動の実行を改善することに関する。

【0002】

【従来の技術】はじめに、本明細書で用いる用語について説明する。本明細書で用いる一部の用語には辞書上の意味も含まれるが、以下のいくつかの用語の用語集が役立つであろう。

【0003】ESA/390は、現在使用されているエンタープライズ・システム体系で使用される、インターナショナル・ビジネス・マシーンズ・コーポレーションのメインフレーム・システム・アーキテクチャを指す。このアーキテクチャは、世界中のメインフレーム・アーキテクチャの標準となっており、既存の命令を使用することが、既存のアーキテクチャの使用を可能にする上で重要である。

【0004】MVCは、既存のESA/390命令の1

11

つであるMOVE CHARACTER (文字移動) の略号である。

【 0 0 0 5 】 NC は、既存の ESA / 3 9 0 命令の 1 つである AND CHARACTER (文字論理積) の略号である。

【 0 0 0 6 】 OC は、既存の ESA / 3 9 0 命令の 1 つである OR CHARACTER (文字論理和) の略号である。

【 0 0 0 7 】 XOR は、既存の ESA / 3 9 0 命令の 1 つである XOR CHARACTER (文字排他的論理和) の略号である。

【 0 0 0 8 】 MVZ は、既存の ESA / 3 9 0 命令の 1 つである MOVE ZONES (ゾーン移動) の略号である。

【 0 0 0 9 】 MVN は、既存の ESA / 3 9 0 命令の 1 つである MOVE NUMERICS (数値移動) の略号である。

【 0 0 1 0 】 SAR1 は、第 1 記憶オペランド・アドレスの略号である。

【 0 0 1 1 】 SAR2 は、第 2 記憶オペランド・アドレスの略号である。

【 0 0 1 2 】 AGEN と AG は、アドレス生成の略語として同じ意味で使用する。

【 0 0 1 3 】 次に、従来の技術について述べる。本発明の背景として、読者がこの議論を速やかに理解し、本発明が従来の技術とどう異なるか理解できるように、IBM テクニカル・ディスクロージャ・ブルテン所載のいくつかの論文を簡単に紹介する。まず、本発明は、MVC 命令、ならびに MVC 命令と同様の形で潜在的にオーバーラップする可能性のあるオペランドを有することのできる命令について、破壊的な記憶オペランドのオーバーラップを検出するための予測機構を提供するものであることを理解されたい。本発明は、ESA / 3 9 0 プロセッサ・アーキテクチャで使用されるタイプのアドレス計算における破壊的なオペランドのオーバーラップの予測に適用される。本発明が、他のアーキテクチャのアドレス指定モードをサポートできるように容易に拡張できることは、想定した 6 4 ビットのアドレス指定モードを含むように検出を拡大することによって実証される。

【 0 0 1 4 】 インターナショナル・ビジネス・マシーンズ・コーポレーション (以下では IBM と称する) は、IBM 社内で行った発明を一般に知らせるため、テクニカル・ディスクロージャ・ブルテン (TDB) と称する雑誌を発行している。TDB で当技術分野に関する論文を探してみると、TDB 1 2 - 8 2 でガイ (Ngai) 等が、破壊的オペランド・オーバーラップ移動命令の実行を扱うための高速アルゴリズムを発表していることに気付く。このアルゴリズムは、MOVE CHARACTER、MOVE NUMERICS、MOVE ZONES などのタイプの命令に適している。しかし、破壊

12

的オペランド・オーバーラップ条件の決定は仮定されており、明記されていない。この仮定条件の予測が本発明の主題である。

【 0 0 1 5 】 ビーチャー (Beetcher) 等は TDB 0 6 - 8 1 で、命令の挙動が、オペランドがオーバーラップしているかどうかに依存する、Move Long Embellished (MLE) 命令を定義している。この命令定義はこの TDB 論文に明記されているが、記憶オペランドのオーバーラップを検出するための機構は定義されていない。

【 0 0 1 6 】 ガイ等は TDB 0 6 - 7 6 で、宛先アドレスとソース・アドレスのビットを使って ROS (書込み専用記憶装置) にアクセスすることにより、破壊的な記憶オペランドのオーバーラップを判定するハードウェア機構を発表している。この ROS は、記憶オペランドのオーバーラップがその組合せから発生する可能性があるかどうかを示す標識を各記憶位置に格納している (過剰指示) 。アドレスの全ビットが使用される。複数のアドレス指定モードのサポートについての説明はない。また、ガイ等の方法では、計算機のデータフロー幅に関する条件が存在するかどうかを判定する。本発明では、アーキテクチャによって示される所によれば記憶オペランドが破壊的なオペランドのオーバーラップを有するかどうかを厳密に予測する。この条件は、オペランド内の任意のバイトがオーバーラップする場合に発生する。トレース解析によれば、現在の高性能プロセッサ実施態様で普通と見られるダブルワードの記憶データ・バス幅よりもオペランド長が短い場合、オペランドがオーバーラップする条件はまれである。したがって、一般的ケースについてオーバーラップを検出することによって、性能は余り低下しない。いずれにせよ、本発明の指示は厳密であり、過剰指示ではない。第 2 に、本発明では、その条件を完全に論理によって検出し、アレイ・アクセスには頼らない。したがって、本発明の決定は、ガイ等が発表した機構よりも高速でセル効率が高くなるはずである。最後に、ガイ等が発表した機構では、本発明のように基底、変位およびオペランド長からではなく、生成された後のアドレスからオーバーラップ条件を予測している。

【 0 0 1 7 】 ブルックス (Brooks) は TDB 0 6 - 7 6 で、データ・フィールドに対する両立できない同時アドレスを防止するためのオペランド整合性アテンダント (OCA) を発表している。この論文で記述されるハードウェア機構では、開始オペランド・アドレスと終了オペランド・アドレスの比較によって、オペランドがオーバーラップするかどうか判定する。したがって、決定の前にオペランドの開始アドレスだけでなく終了アドレスも計算しておく必要がある。この開始アドレスと終了アドレスが得られた後に、オペランドのオーバーラップが存在することを近似して、比較の速度を上げる機構が利用できる。一方、本発明では、ソース・アドレス及び宛

13

先アドレス(上記TDB論文の開始アドレス)の生成と並行して、記憶オペランド・オーバーラップ条件を判定する。さらに、本発明の指示は厳密である。

【0018】ベーカー(Baker)等はTDB 01-82で、オペランドがオーバーラップしたとき、オペランドがバイトごとに処理されるかのようにその挙動が進行する、多くの命令を実行するためのアルゴリズムを発表した。オペランド・オーバーラップの検出が既知であり、実行のためのアルゴリズムが準備されていると仮定されている。本発明の主題である、オペランド・オーバーラップ条件を検出するための機構は示されていない。

【0019】

【発明が解決しようとする課題】本発明は、2つの記憶オペランドが破壊的にオーバーラップするかどうか予測するために、記憶アドレスの生成と並行して実行する、特殊なハードウェアの使用に関する。記憶装置間命令の実行の挙動は、その命令が記憶オペランドに対してバイトごとに実行されるかのように進むようなアーキテクチャになっていることがしばしばである。これは、これらすべての動作をバイトごとに実行することによって達成できる。しかし、ソース・オペランドと宛先オペランドが破壊的にオーバーラップしない場合は、複数のバイトに対して並列に動作を実行し、なおかつそれがバイトごとの実行のように見えるようにすることができる。したがって、記憶オペランドが破壊的にオーバーラップすることを検出し、そうでない場合は一般に複数のバイトに対して実行を進めさせ、オペランドが破壊的にオーバーラップする場合は動作をバイトごとに実行することにより、命令の実行速度を高めることができる。アーキテクチャ上の要件を満たすには、取り出したデータが記憶される前に、破壊的オペランド・オーバーラップ条件がわかっているなければならない。場合によっては、記憶オペランドのオーバーラップが判定されている間に有用な作業を実施することができるが、頻繁に使用されるMOVE CHARACTER (MVC) 命令では、このギャップを埋める有用な作業はない。したがって、オペランド・オーバーラップ条件が判定されるまで、実行を停止しなければならない。

【0020】

【課題を解決するための手段】従来、破壊的記憶オペランド・オーバーラップを検出するには、まずソース・オペランドと宛先オペランドの記憶アドレスを計算し、続いて両記憶アドレスを比較して、次式が成立するかどうか判定していた。

【数14】 $0 \leq SAR1 - SAR2 \leq L$

【0021】上式で、SAR1とSAR2は、それぞれ宛先オペランドとソース・オペランドの記憶オペランド・アドレスを表し、Lはオペランドの長さを表す。パイプライン式プロセッサでは、この検出に、記憶アドレスの生成に加えて少なくとももう1サイクルを要した。し

14

たがって、MVC命令が進行できるようになる前に少なくとも1停止サイクルが必要である。本発明によると、MVC命令のソース・オペランドと宛先オペランドの基底と変位、および命令中で指定されるオペランドの長さから直接にMVC命令の記憶オペランド・オーバーラップ条件を判定することができる。この判定をソース記憶アドレス及び宛先記憶アドレスの生成と並行して実行することにより、頻繁に呼び出されるMVC命令の実行中に停止サイクルが必要でなくなる。MVC命令について1停止サイクル節約されるだけで、オペランド長が8バイト未満の比較的良好に見られるケース(84%)ではMVC命令の実行速度が21%向上し、オペランド長が8バイトを超えるケース(16%)では9%向上する。MVC命令の頻度により、上記の数字は、代表的命令混合体でCPIの純減として反映され、MVC命令だけの場合はそれが0.02CPIとなり、しかも最小のハードウェアを追加するだけで達成される。本発明は、ESA/390プロセッサ用に設計されており、そのアーキテクチャにとって既知のすべてのアドレス指定モードについて破壊的記憶オペランドを予測することができる。また想定される64ビットのアドレス指定モードをサポートするように予測機構を拡張することにより、本発明を他のアーキテクチャに拡張できることも実証されている。

【0022】上記その他改善点は、下記の詳しい説明に示してある。本発明とその利点および特徴をよりよく理解するために、以下の説明および図面を理解されたい。

【0023】

【実施例】本発明の好ましい実施例を詳細に考察する前に、ソース・アドレスと宛先アドレスの破壊的オーバーラップを予測するのに使用される条件を展開しておくのが有意義であろう。

【0024】IBM ESA/390などのアーキテクチャでは、オーバーラップする記憶オペランドに対する命令の実行をアーキテクチャ方式で実施することが必要な場合がある。それには、命令の実行を開始する前にそのようなオーバーラップ条件の検出が必要となることがあり得る。これらのタイプの命令のいくつか(主な例はESA/390におけるMOVE CHARACTER (MVC) 命令)の頻度と、その命令の大部分の呼出しの際にその実行に必要なサイクル数とが与えられているものとする、このプロセスをオペランド・オーバーラップの順次検出によって実施する際には、大きな不利益を受ける恐れがある。このことは、オペランド・オーバーラップの速やかな判定が必要なことを示唆している。前記のSS命令MVC、およびESA/390アーキテクチャで見られる他のSS命令AND CHARACTER (NC)、OR CHARACTER (OC)、XOR CHARACTER (XC)、MOVE ZONES (MVZ)、MOVENUMERICS (MVN)

15

では、オペランドのオーバーラップする条件は次式で表すことができる。

【数15】 $0 \leq SAR1 - SAR2 \leq L$

【0025】上式で、 $SAR1$ と $SAR2$ は、それぞれ宛先オペランドとソース・オペランドの記憶アドレスを表し、 L はオペランドの長さを表す。上記のリスト中の命令のうちでMVCが命令ストリーム中で最も頻繁に発生する。したがって、上記のオーバーラップを、MVC型オーバーラップと呼ぶことにする。さらに、他のES *

$$SAR2 \leq SAR1 + L1 \leq SAR2 + (2 * L2) - 2$$

【数18】

$$SAR2 \leq SAR1 + L1 \leq SAR2 + 2 * 55$$

【0026】MVC型オーバーラップは厳密に検出しなければならないが、後者の命令群でのオーバーラップは過剰指示することができる。オーバーラップの過剰指示は、PACK、ZAPおよびUNPKでは数式19によって実施でき、TRとTRTでは数式20によって実施できる。

【数19】 $|SAR1 - SAR2| \leq 15$

【数20】 $|SAR1 - SAR2| \leq 2 * 55$

【0027】前述のように、MVCは、代表的な命令混合体で最も頻繁に現れる命令の1つである。したがって、設計が与えられている場合、オーバーラップの検出に使用される機構が、命令当りサイクル数(CPI)に影響を与えることがあり得る。従来技術では、MVCオーバーラップ検出は、両方の記憶オペランド・アドレス $SAR1$ と $SAR2$ を計算し、続いて $SAR1$ を $SAR2$ と、また $SAR1 - SAR2$ を L と比較することからなっていた。このような実施態様は記憶バスの過小利用をもたらし、性能上のボトルネックを生じる可能性がある。というのは、 $SAR1$ と $SAR2$ の計算および $SAR1$ と $SAR2$ の比較にはAGEN ALUが必要だからである。オーバーラップの検出から生じ得る性能低下の例として、単一サイクル命令の場合は

ID-AG-EX

記憶アクセスを含む2サイクル命令の場合は

ID-AG-CA

μF -AG-EX

のパイプライン構造を使用する、高性能スーパー scaler・プロセッサ用のPU設計を考えてみる。上式で μF はマイクロワード取出しを表す。このパイプライン中で、IDは命令復号、AGはアドレス生成、CAはデータ・キャッシュ・アクセス、EXは命令実行を表す。命令のプット・アウェイ(put-away)は隠されている。すなわち、実行の結果は、次の実行サイクルで使用される必要な機能ユニットへの入力としてバイパスさせることができる。PUは、最大3個のESA/390命令を同時に実行できる能力をもつ設計になっているものとする。この能力により、2つのアドレス生成ALUがデー

16

*A/390命令でもオーバーラップは起こり得る。そうした命令としては、数式16が成立するときにオーバーラップが発生するPACKやZAP、数式17が成立するときにオーバーラップが発生するUNPK、および数式18が成立するときにオーバーラップが発生するTRとTRTがある

【数16】

$$SAR2 \leq SAR1 + L1 < SAR2 + L2$$

【数17】

タフロー中に含まれることになる。ここでは、MVCのようなSS命令では、MVC命令の復号に続くAGサイクル中にソース記憶アドレスと宛先記憶アドレスが同時に生成され得ることを示唆しておく。プロセッサのすべての命令はマイクロコードで制御され、単一サイクル命令は1つのマイクロ命令を呼び出し、複数サイクル命令はマイクロルーチンを呼び出す。MVCは複数サイクル命令の範疇に含まれ、したがってその実行中に複数のマイクロ命令が呼び出される。このようなルーチンでは、実行の結果、次のパイプライン・シーケンスが発生する。

第1 マイクロワード ID-AG-CA

第2 マイクロワード μF -AG-EX

第3 マイクロワード μF -AG-EX

上式で μF はマイクロワードの取出しを表し、命令の実行が完了するまでこのシーケンスが反復される。次に、本発明のオペランド・オーバーラップ予測装置を記述するのに使用するMVC命令の可能な実施態様について述べる。この実施態様では、第1のマイクロ命令が、第1オペランドでアドレス指定される記憶装置からのロードを指定する。この第1マイクロ命令のAGサイクル中に両方のオペランド・アドレスが2つのAGEN ALUによって計算され、オペランド1の取出しコマンドが形成され、データ・キャッシュに送られる。次のサイクルで、データ・キャッシュから取り出されたデータがデータ・バス上に置かれ、プロセッサに受諾される。プロセッサはキャッシュからデータ・バスに取り出されたデータをバイパスすることができるので、このサイクル中に取り出されたデータをデータ・キャッシュ中に格納するために記憶コマンドを形成することができると考えられる。しかし、アーキテクチャを遵守するには、データを記憶する前にオペランド・オーバーラップがわかっている必要がある。従来技術では、第2のマイクロ命令のAGサイクル中でのオーバーラップを判定するため、第1命令のAGサイクル中に計算される記憶オペランド・アドレス $SAR1$ と $SAR2$ を次のサイクルでオーバーラップ検出論理機構にバイパスしなければならない。次にこのオーバーラップ検出の結果を分岐条件として使って、アーキテクチャに合致する形でMVCを実行するには2つのマイクロルーチンのどちらにアクセスすべきか

20

30

40

50

17

を判定する。その結果、従来技術では第2のマイクロ命令が実際にNOPとなり、取り出されたデータを格納するために適当なマイクロルーチンにアクセスする前に、オーバーラップを判定することが可能となる。したがって、記憶バスは過小利用となる。第1マイクロワードのAGサイクル中にオーバーラップ条件を予測し、第2のマイクロワードを取り出す際に適当な経路指定を選択するための分岐条件としてそれを利用できるようにすることができれば、この過小利用を軽減することができる。以下に、2つの記憶オペランド・アドレスSAR1とSAR2の計算と並行してオーバーラップ条件を予測できる、本発明を提示する。このようにすると、MVC型オーバーラップの検出を必要とする命令を呼び出すごとに1サイクルが節約できる。代表的命令混合体を使用すると、オペランド長が8以下のMVCが呼び出される頻繁なケース(84%)ではMVC命令を実行するのに要するサイクル数が、平均4.7サイクルから平均3.7サイクルに減少し、21%の改善が得られる。MVCのオペランド長が8を越える比較的なめなケース(16%)では、実行サイクル数は11.8サイクルから10.8サイクルに減少し、9%の改善となる。CPIの純減少率は0.02であり、それが最小の追加ハードウェアで達成される。

【0028】オーバーラップをもたすアドレス条件は5つのクラスに分けることができる。これらのケースを図1に示す。第1のケースでは、ソース・オペランドが高位記憶域と低位記憶域の境界を越えて折り返し(ラップ; wrap)、低位記憶域にある宛先オペランド・アドレスとオーバーラップするために、オーバーラップが発生する。第2のケースでは、ソース・オペランドと宛先オペランドが共に高位記憶域から低位記憶域に折り返す。この場合、宛先オペランドはソース・オペランドより上方にあり、低位アドレス位置に続いている。第3のケースでは、ソース・アドレスと宛先アドレスが記憶域内の同じ256バイト・ブロックで開始し、ソース・オペランドの全体がこの256バイト・ブロック内にある。第4のケースでは、両方のオペランドが記憶域の真中の同じ256バイト・ブロック内で開始し、両方のオペランドが共に次の順次256バイト・ブロック中にあふれ出る(スピル)。最後に第5のケースでは、宛先オペランドがソース・オペランドの次の順次ブロックで開始する。ソース・オペランドはこの順次ブロック中にあふれ出して、宛先オペランドとオーバーラップする。オーバーラップしないケースは、以上のオーバーラップするケースに含まれない、残りのすべての場合を含む。それらのケースを図2および3に示す。次のこのオーバーラップするケースとオーバーラップしないケースのすべてを使って、ESA/390アーキテクチャでサポートされるすべてのアドレス指定モードでオーバーラップを検出するための本発明を開始する。このモードには、24ビ

18

ットおよび31ビット・アクセス・レジスタ・モードが含まれる。さらに、64ビットの基底と12ビットの変位のアドレス計算も仮定する。この24ビットと31ビットのアドレス指定モードが含まれるのは、それらがESA/390アーキテクチャで提供される24ビットおよび31ビット・アクセス・レジスタ・モードのサブセットと見なせるからでもある。図1のケース1やケースのような折返しの場合は、64ビットの基底と12ビットの変位の加算の場合、64ビット・アドレスのMSBが有効アドレスをもつためには0でなければならない、かつ折返しは $2^{64}-1$ アドレス境界で発生すると定義されているものと仮定することにより、考察から除外する。したがって、このアドレス指定モードでは、メモリ・アドレス指定が仮定アドレス空間の最高位256バイト・ブロックで発生し得ない。以下の考察では、2の補数を大文字、たとえばAで表すことにする。2の補数は、m個のビット a_i からなる。ただし、 $0 \leq i \leq m-1$ であり、 a_0 は最上位ビット(MSB)を表し、 a_{m-1} は最下位ビット(LSB)を表す。さらに、表記A($i:j$)(ただし、 $i \leq j \leq m-1$)を使って、A内部にあってiとjの間にあり両端を含むビットから構成される数を表すことにする。この表記を使用して、以下の定理を、MVC型オーバーラップを検出するための本発明の基礎として証明する。

【0029】定理1 2の補数演算を使って、2つの暗示的な正の2の補数の減算 $A-B$ を行うと、 $A \geq B$ の場合、 $A=0 \parallel A(i:m-1)$ および $B=0 \parallel B(i:m-1)$ として、ビット位置iからすぐ下のビット位置i-1へのキャリーが発生する。

【0030】証明: $A(i:m-1) > B(i:m-1)$ の場合、 $a_j=1$ かつ $b_j=0$ であり、 $i \leq p < j$ であるすべてのpについて $a_p=b_p$ となるjが存在する。2の補数の減算 $A-B$ を実行するには、Bの1の補数にAを加え、ホット1を位置m-1に与える。しかし、Bは1の補数化されているので、 $\overline{b_p} = \overline{a_p}$ かつ $a_j=1$ および $\overline{b_j}=1$ である。ただし、

【数21】

$$\overline{b_p}$$

を $\overline{b_p}$ と表記する。他の文字にバーが付与される場合も、これに準ずる。したがって、 $A-B$ はjからj-1へのキャリーが発生する。しかし、位置j-1では、 $\overline{b_j} = \overline{a_j}$ であるため、 $a_{j-1}=1$ かつ $\overline{b_{j-1}}=0$ 、または $a_{j-1}=0$ かつ $\overline{b_{j-1}}=1$ である。したがって、j-1へのキャリーを a_{j-1} および $\overline{b_{j-1}}$ に加えると、j-2へのキャリーが発生する。 $i+1 \leq q \leq j-2$ である各ビット位置qについてj-1の場合と同じ考察を適用すると、ビット位置iで、iへのキャリーは1であり、 a_i と b_i のどちらか一方だけが1となる。したがって、ビット位置iからi-1へのキャリーが発生することになる。

19

【 0 0 3 1 】 $A(i : m-1) = B(i : m-1)$ の場合、 $i \leq j \leq m-1$ であるすべての j について $a_j = b_j$ である。B は加算の前に補数化されるので、各ビット位置 j はすべて0 に加算される1 からのなる。したがってホット1 がLSB 位置に与えられるので、 $m-1$ でキャリーが発生する。このキャリーが位置 $m-2$ に伝播し、そこで $m-3$ へのキャリーが発生する。このキャリーがあらゆるビット位置 j に伝播し、ビット i から $i-1$ へのキャリーが発生する。証明終り。

【 0 0 3 2 】 定理2 2 の補数演算を使って、2 つの暗示的な正の2 の補数の減算 $A-B$ を行うと、 $A < B$ の場合、 $A=0 \parallel A(i : m-1)$ および $B=0 \parallel B(i : m-1)$ として、ビット位置 i から次のビット位置 $i-1$ へのキャリーは発生しない。

【 0 0 3 3 】 証明: $A(i : m-1) < B(i : m-1)$ の場合、 $a_j = 0$ かつ $b_j = 1$ であり、 $i \leq p < j$ となるすべての p について $a_p = b_p$ となる j が存在する。2 の補数の減算 $A-B$ を実行するには、B の1 の補数にAを加え、ホット1 を位置 $m-1$ に与える。しかし、B は1 の補数化されているので、 $b_p = \text{バー}a_p$ かつ $a_j = 0$ および $\text{バー}b_j = 0$ である。 a_j と $\text{バー}b_j$ が共に0 なので、位置 j へのキャリーである c_{j+1} の値がどうであれ、 j から $j-1$ へのキャリーは0 となる。したがって、 $A-B$ は、 j から $j-1$ へのキャリーが発生しない。位置 $j-1$ では、 $\text{バー}b_j = \text{バー}a_j$ であるため、 $a_{j-1} = 1$ かつ $\text{バー}b_{j-1} = 0$ または $a_{j-1} = 0$ かつ $\text{バー}b_{j-1} = 1$ である。したがって、 $j-1$ へのキャリーがないことは、 a_{j-1} と $\text{バー}b_{j-1}$ を加算して $j-1$ へのキャリー c_{j-1} が発生しても、 $j-2$ へのキャリーは発生し

20

ないことを意味している。 $i+1 \leq q \leq j-2$ である各ビット位置 q について $j-1$ の場合と同じ考察を適用すると、ビット位置 i で、 i へのキャリーは0 であり、 a_i と $\text{バー}b_i$ のどちらか一方だけが1 となる。したがって、ビット i から $i-1$ へのキャリーは発生しない。証明終り。

【 0 0 3 4 】 MVC 型オペランド・オーバーラップを検出するための本発明を引き続き展開するため、以下の表記法を利用する。まず、SAR2 と SAR1 をそれぞれ A と B で表す。両者はそれぞれ a_i と b_i で表される個々のビットを有する。ただし、 $0 \leq i \leq 63$ である。8 ビット・オペランド長を OL (0 : 7) で表し、左に56 個の0 を連結したものを L で表す。したがって、L i は、 $0 \leq i \leq 55$ の場合は0、 $56 \leq i \leq 63$ の場合は OL (0 : 7) となる。以下の議論では、A、B、L の以下の加算/減算の組合せが必要となる。その組合せならびにその表記法は、

1. $A-B$ これを Δ で表し、個々のビットを δ_i で表し、あるビット位置から次のビット位置へのキャリーを κ_i で表す。
2. $A+L$ これを E で表し、個々のビットを ϵ_i で表し、キャリーを γ_i で表す。
3. $\Delta+L$ これを S で表し、個々のビットを S_i で表し、キャリーを c_i で表す。
4. $E-B$ これを S^+ で表し、個々のビットを S_i^+ で表し、キャリーを λ_i で表す。

この表記法を表1 に要約して示す。

【 表 1 】

MVC 型オーバーラップ検出装置用の表記法				
エンティティ	表現	文字表現	ビット表現	キャリー表現
SAR2	—	A	a_i	—
SAR1	—	B	b_i	—
SAR2-SAR1	$A-B$	Δ	δ_i	κ_i
SAR2+L	$A+L$	E	ϵ_i	γ_i
(SAR2-SAR1)+L	$\Delta+L$	S	S_i	c_i
(SAR2+L)-SAR1	$E-B$	S^+	S_i^+	λ_i

【 0 0 3 5 】 上記の表記法を用いると、以下の定理が成立する。

【 0 0 3 6 】 定理3 記憶オペランド1 および2 がMV C型オーバーラップを有する場合、SAR2 がアドレス空間の高位2 5 6 バイト・ブロック内にあり、SAR1 が低位2 5 6 バイト・ブロック内にあるときに、SAR1 がSAR2 より上にマップされるならば、 $0 \leq i \leq 5$ *

*5 の場合 $\delta_i = 1$ である。

【 0 0 3 7 】 証明: ESA/390 アーキテクチャで提供される3 つのアドレス指定モードのそれぞれについて、図1 に示した各ケースごとにこの定理を証明しなければならない。各アドレス指定モードの場合の記憶アドレスの表現を表2 に示す。

【 表2 】

ESA/390 および想定した64ビット・アドレス指定モードでの アドレス表現	
モード	アドレス表現
24ビット・アクセス・レジスタ (AR24ビット) モード	$a_0 a_1 a_{31} 0 0 \cdots 0 a_{40} a_{41}$ $\cdots a_{55} a_{56} a_{57} \cdots a_{63}$
31ビット・アクセス・レジスタ (AR31ビット) モード	$a_0 a_1 \cdots a_{31} 0 a_{33} \cdots a_{55} a_{56}$ $a_{57} \cdots a_{63}$
64ビット・アドレス指定	$0 a_1 \cdots a_{55} a_{56} a_{57} \cdots a_{63}$

【 0 0 3 8 】 表2 では、0 であることがわかっているSAR2 のビット位置は0 で示してあり、1 と0 のどちらかであるビット位置は、SAR2 を先にA で表したので添字付きのa で表してある。添字は変数中のビット位置を表す。同様に、a の代りにb を使ってSAR1 を表すことができる。この表現を使つて、図1 に示した各ケースについて考察しなければならない。SAR2 がアドレス空間の高位2 5 6 バイト・ブロック内にあり、SAR1 が低位2 5 6 バイト・ブロック内にある場合では、アドレスを1 ビット位置拡張し、そのビット位置を1 にすることにより、SAR1 をSAR2 より上にマップすることができる。したがって、24ビット・アドレス空間の場合、そのアドレスのMSBに25番目のビットが連結され、それが1 にされることになる。表2 に示したアドレス表現では、各アドレス指定モードについてアドレス空間がすでに拡張されており、表に示すようにビット位置が0 である。したがって、この表現でSAR1 をSAR2 より上にマップするには、次のビット位置を1 にするだけでよい。たとえば、アドレスの24ビットは、アドレスの少なくとも24ビット位置を必要とするはずであり、これは表2 の表現ではビット40 ~63 に対応する。24ビット・アドレス指定では、SAR1 をSAR2 より上にマップするために、ビット39 を0 にする。

同様に31ビット・アドレス指定では、SAR1 をSAR2 より上にマップするために、アドレスのビット32 を1 にする。

【 0 0 3 9 】 定理3 ケース1 ケース1 は、A がアドレス空間の高位2 5 6 バイト・ブロック内にあり、B が低位2 5 6 バイト・ブロック内にあり、A の最上位ビットがB の最上位ビットに等しいことを特徴とする。A がアドレス空間の高位2 5 6 バイト・ブロック内にあり、B が低位2 5 6 バイト・ブロック内にあるため、B はA より上にマップされるはずである。さらに、 $B(56:63) \leq E(56:63)$ かつ $E(56:63) < A(56:63)$ なので、 $A(56:63) > B(56:63)$ である。したがって、定理1 により、 $\kappa_{56} = 1$ となる。以下で3 つのアドレス指定モードのすべてを考慮する。

【 0 0 4 0 】 AR24ビット・モード このアドレス指定モードでは、B をA より上にマップするために、ビット39 を論理1 にする。ビット39 より上位のビット位置は等しいので、 $i \leq 38$ であるi について $a_i = b_i$ である。その結果、 $A - B$ は数式22 で表すことができ、ビット1 が暗黙のうちに強制的にビット63 に入れられる。

【 数22 】

$$\begin{array}{c}
 23 \qquad \qquad \qquad 24 \\
 a_0 \ a_1 \dots a_{31} 00 \dots 011 \dots 1 a_{56} \ a_{57} \dots a_{63} \\
 \overline{a_0} \ \overline{a_1} \dots \overline{a_{31}} 11 \dots 011 \dots 1 \overline{b_{56}} \ \overline{b_{57}} \dots \overline{b_{63}}
 \end{array}$$

【0041】 $\kappa_{56}=1$ であり、かつ $40 \leq i \leq 55$ である各ビット位置がその位置へのキャリーに加算すべき1を2個を有するので、 $40 \leq i \leq 55$ であるすべての i について、 $\delta_i=1$ かつ $\kappa_i=1$ となる。ビット位置39では2個の0が $\kappa_{40}=1$ と加算され、したがって $\sigma_{39}=1$ かつ $\kappa_{39}=0$ である。 $0 \leq i \leq 38$ である各ビット位置では、1とキャリーの0に0が加算される。したがって、 $0 \leq i \leq 38$ であるすべての i について $\delta_i=1$ かつ $\kappa_i=0$ であり、したがって $0 \leq i \leq 55$ であるすべての i について $\delta_i=1$ である。

【0042】AR31ビット・モード このアドレス指定モードでは、BをAより上にマップするために、ビット32を論理1にする。ビット32より上位のビット位置は等しいので、 $i \leq 31$ である i について $a_i=b_i$ である。その結果、 $A-B$ は数式23で表すことができ、ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【数23】

$$a_0 \ a_1 \dots a_{31} 01 \dots 1 a_{56} \ a_{57} \dots a_{63}$$

$$\overline{a_0} \ \overline{a_1} \dots \overline{a_{31}} 01 \dots 1 \overline{b_{56}} \ \overline{b_{57}} \dots \overline{b_{63}}$$

【0043】24ビット・アドレス指定の場合と同様に、 $\kappa_{56}=1$ であり、かつ $33 \leq i \leq 55$ である各ビット

$$a_0 \ a_1 \dots a_{31} 00 \dots 011 \dots 1 a_{56} \ a_{57} \dots a_{63}$$

$$\overline{a_0} \ \overline{a_1} \dots \overline{a_{31}} 11 \dots 100 \dots 0 \overline{b_{56}} \ \overline{b_{57}} \dots \overline{b_{63}}$$

【0047】 $\kappa_{56}=0$ であり、かつ $0 \leq i \leq 55$ である各ビット位置がそのビット位置へのキャリーに加算すべき1個の1と1個の0を有するので、各ビットでの結果は、 $\delta_i=1$ かつ $\kappa_i=0$ である。したがって、 $0 \leq i \leq 55$ であるすべての i について $\delta_i=1$ である。

【0048】AR31ビット・モード ビット33より上位のビット位置がBとAで等しく、したがって $i \leq 32$ である i について $a_i=b_i$ である。その結果、 $A-B$ は数式25であらわすことができ、ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【数25】

$$a_0 \ a_1 \dots a_{31} 01 \dots 1 a_{56} \ a_{57} \dots a_{63}$$

$$\overline{a_0} \ \overline{a_1} \dots \overline{a_{31}} 10 \dots 0 \overline{b_{56}} \ \overline{b_{57}} \dots \overline{b_{63}}$$

【0049】 $\kappa_{56}=0$ であり、かつ $0 \leq i \leq 55$ である各ビット位置が0及びそのビット位置へのキャリーに加

算すべき1個の1を有するので、 $\delta_i=1$ かつ $\kappa_i=0$ である。したがって、 $0 \leq i \leq 55$ であるすべての i について $\delta_i=1$ である。ビット位置32では2個の0が $\kappa_{33}=1$ と加算され、したがって $\delta_{32}=1$ かつ $\kappa_{32}=0$ である。 $0 \leq i \leq 31$ である各ビット位置については、1とキャリーの0に0が加算される。したがって、 $0 \leq i \leq 31$ であるすべての i について $\delta_i=1$ かつ $\kappa_i=0$ であり、したがって $0 \leq i \leq 55$ であるすべての i について $\delta_i=1$ である。

【0044】64ビット・アドレス指定 64ビット・アドレス指定ではケース1は生じ得ない。その結果、64ビット・アドレス指定の場合は、このケースに対処するためにBをAより上にマップする必要はない。

【0045】定理3 ケース2 ケース2は、AとBが共にアドレス空間の高位256バイト・ブロック内にあり、 $A(56:63) < B(56:63)$ であることを特徴とする。したがって、定理2より、 $\kappa_{56}=0$ となる。

【0046】AR24ビット・モード ビット40より上位のビット位置がBとAで等しく、したがって $i \leq 39$ である i について $a_i=b_i$ である。その結果、 $A-B$ は数式24で表すことができ、ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【数24】

算すべき1個の1を有するので、 $\delta_i=1$ かつ $\kappa_i=0$ である。したがって、 $0 \leq i \leq 55$ であるすべての i について $\delta_i=1$ である。

【0050】64ビット・アドレス指定 64ビット・アドレス指定ではケース2は生じ得ない。

【0051】定理3 ケース3および4 これらのケースは、BとAが共にアドレス空間の同一の256バイト・ブロック内にあり、 $A(56:63) < B(56:63)$ であることを特徴とする。定理2により、 $\kappa_{56}=0$ となる。BとAの両方が存在する256バイト・ブロックは任意である。

【0052】AR24ビット・モード ビット40より上位のビット位置がBとAで等しく、したがって $i \leq 39$ である i について $a_i=b_i$ である。その結果、 $A-B$ は数式26で表すことができ、ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【数26】

$$\begin{array}{ccccccc}
 & 25 & & & & & 26 \\
 a_0 & a_1 \dots a_{31} & 0 & 0 \dots 0 & a_{40} & a_{41} \dots a_{55} & a_{56} & a_{57} \dots a_{63} \\
 \hline
 \overline{a_0} & \overline{a_1} \dots \overline{a_{31}} & 1 & 1 \dots 1 & \overline{a_{40}} & \overline{a_{41}} \dots \overline{a_{55}} & \overline{a_{56}} & \overline{a_{57}} \dots \overline{b_{63}}
 \end{array}$$

【0053】 $\kappa_{56}=0$ であり、かつ $0 \leq i \leq 55$ である各ビット位置が0及びそのビット位置へのキャリーに計算すべき1個の1を有するので、 $\delta_i=1$ かつ $\kappa_i=0$ である。したがって、 $0 \leq i \leq 55$ であるすべての*i*について $\delta_i=1$ である。

【0054】AR31ビット・モード ビット33より*10

$$\begin{array}{ccccccc}
 a_0 & a_1 \dots a_{31} & 0 & a_{33} \dots a_{55} & a_{56} & a_{57} \dots a_{63} \\
 \hline
 \overline{a_0} & \overline{a_1} \dots \overline{a_{31}} & 1 & \overline{a_{33}} \dots \overline{a_{55}} & \overline{b_{56}} & \overline{b_{57}} \dots \overline{b_{63}}
 \end{array}$$

【0055】 $\kappa_{56}=0$ であり、かつ $0 \leq i \leq 55$ である各ビット位置が0及びそのビット位置へのキャリーに計算すべき1個の1を有するので、 $\delta_i=1$ かつ $\kappa_i=0$ である。したがって、 $0 \leq i \leq 55$ であるすべての*i*について $\delta_i=1$ である。

【0056】64ビット・アドレス指定 このケースで20は64ビット・アドレス指定の場合、A-Bは次式で表すことができ、ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【数28】

$$\begin{array}{ccccccc}
 0 & a_1 \dots a_{55} & a_{56} & a_{57} \dots a_{63} \\
 1 & \overline{a_1} \dots \overline{a_{55}} & \overline{b_{56}} & \overline{b_{57}} \dots \overline{b_{63}}
 \end{array}$$

【0057】 $\kappa_{56}=0$ であり、かつ $0 \leq i \leq 55$ である各ビット位置が0及びそのビット位置へのキャリーと加30算すべき1個の1を有するので、 $\delta_i=0$ かつ $\kappa_i=0$ で※

$$\begin{array}{ccccccc}
 a_0 & a_1 \dots a_{31} & 0 & 0 \dots 0 & a_{40} & a_{41} \dots a_{j-1} & 0 & 1 & 1 \dots 1 & a_{56} & a_{57} \dots a_{63} \\
 \hline
 \overline{a_0} & \overline{a_1} \dots \overline{a_{31}} & 1 & 1 \dots 1 & \overline{a_{40}} & \overline{a_{41}} \dots \overline{a_{j-1}} & 0 & 1 & 1 \dots 1 & \overline{b_{56}} & \overline{b_{57}} \dots \overline{b_{63}}
 \end{array}$$

【0060】 $\kappa_{56}=1$ であり、かつ $j+1 \leq i \leq 55$ である各ビット位置がそのビット位置へのキャリーと加算すべき1を2個有するので、 $j+1 \leq i \leq 55$ である*i*について $\delta_i=1$ かつ $\kappa_i=1$ である。ビット位置*j*では a_j と b_j が共に0であり、かつ $\kappa_{j+1}=1$ である。したがって、 $\delta_j=1$ かつ $\kappa_j=0$ である。0と*j-1*の間の各ビット位置では、加数のうちただ1個が1であり、そのビット位置へのキャリーは0である。したがって、 $0 \leq i \leq j-1$ である*i*について $\delta_i=1$ であり、したがって $0 \leq i \leq 55$ であるすべての*i*について $\delta_i=1$ で★

$$\begin{array}{ccccccc}
 a_0 & a_1 \dots a_{31} & 0 & a_{33} \dots a_{j-1} & 0 & 1 & 1 \dots 1 & a_{56} & a_{57} \dots a_{63} \\
 \hline
 \overline{a_0} & \overline{a_1} \dots \overline{a_{31}} & 1 & \overline{a_{33}} \dots \overline{a_{j-1}} & 0 & 1 & 1 \dots 1 & \overline{b_{56}} & \overline{b_{57}} \dots \overline{b_{63}}
 \end{array}$$

【0062】 $\kappa_{56}=1$ であり、かつ $j+1 \leq i \leq 55$ で50ある各ビット位置がそのビット位置へのキャリーと加算

*上位のビット位置がBとAで等しく、したがって $i \leq 32$ である*i*について $a_i=b_i$ である。その結果、A-Bは数式27で表すことができ、ホット1が暗黙のうちに強制的にビット63に入れられる。

【数27】

※ある。したがって、 $0 \leq i \leq 55$ であるすべての*i*について $\delta_i=1$ である。

【0058】定理3 ケース5 このケースでは、Bは、Aが存在する256バイト・ブロックのすぐ上の256バイト・ブロック中にある。さらに、 $B(56:63) \leq E(56:63)$ かつ $E(56:63) < A(56:63)$ である。したがって、 $A(56:63) > B(56:63)$ なので、定理1により $\kappa_{56}=1$ である。

【0059】AR24ビット・モード Bが、Aの存在する256バイト・ブロックのすぐ上の256バイト・ブロック中にあるので、 $40 \leq j \leq 55$ であって、 $a_j=0$ かつ $b_j=1$ であり、 $j < p \leq 55$ である*p*について $a_p=1$ かつ $b_p=0$ となる位置*j*が存在する。さらに、 $i < j$ の場合、 $a_i=b_i$ である。その結果、A-Bは数式29で表すことができ、ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【数29】

★ある。

【0061】AR31ビット・モード Bが、Aが存在する256バイト・ブロックのすぐ上の256バイト・ブロック中にあるので、 $33 \leq j \leq 55$ であって、 $a_j=0$ かつ $b_j=1$ であり、 $j < p \leq 55$ である*p*について $a_p=1$ かつ $b_p=0$ となる位置*j*が存在する。さらに、 $i < j$ の場合、 $a_i=b_i$ である。その結果、A-Bは数式30で表すことができ、ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【数30】

27

すべき1を2個有するので、 $j+1 \leq i \leq 55$ であるすべての i について $\delta_i=1$ かつ $\kappa_i=1$ である。ビット位置 j では、 a_j と b_j が共に0であり、 $\kappa_{j+1}=1$ である。したがって、 $\delta_j=1$ かつ $\kappa_j=0$ である。0と $j-1$ の間の各ビット位置では、加数のうちの1個だけが1であり、そのビット位置へのキャリーは0である。したがって $0 \leq i \leq j-1$ である i について、 $\delta_i=1$ であり、したがって $0 \leq i \leq 55$ であるすべての i について $\delta_i=1$ である。

$$0 \ a_1 \dots a_{j-1} \ 011\dots1 \ a_{56} \ a_{57} \dots a_{63}$$

$$1 \ \overline{a_1} \dots \overline{a_{j-1}} \ 011\dots1 \ \overline{b_{56}} \ \overline{b_{57}} \dots \overline{b_{63}}$$

【0064】 $\kappa_{56}=1$ であり、かつ $j+1 \leq i \leq 55$ である各ビット位置がそのビット位置へのキャリーと加算すべき1を2個有するので、 $j+1 \leq i \leq 55$ であるすべての i について $\delta_i=1$ かつ $\kappa_i=1$ である。ビット位置 j では、 $a_j=b_j$ が共に0であり、 $\kappa_{j+1}=1$ である。したがって、 $\delta_j=1$ かつ $\kappa_j=0$ である。0と $j-1$ の間の各ビット位置では、加数のうちの1個だけが1であり、そのビット位置へのキャリーは0である。したがって、 $0 \leq i \leq j-1$ である i について $\delta_i=1$ であり、したがって $0 \leq i \leq 55$ であるすべての i について $\delta_i=1$ である。証明終り。

【0065】定理4 記憶オペランド1及び2がMVC型オーバーラップを有せず、SAR2がアドレス空間の高位256バイト・ブロック内にありSAR1が低位256バイト・ブロック内にあるときSAR1がSAR2より上にマップされる場合、減算SAR2-SAR1の結果 δ_i は、非オーバーラップ・ケース1、6、11、ビット56からビット55へのキャリーが生成される($\kappa_{56}=1$)とき、及び非オーバーラップ・ケース3を除き $0 \leq i \leq 55$ の場合すべて1ではない。

【0066】定理4 ケース1 ※

$$a_0 \ a_1 \dots a_{31} \ 00\dots011\dots1 \ a_{56} \ a_{57} \dots a_{63}$$

$$\overline{a_0} \ \overline{a_1} \dots \overline{a_{31}} \ 11\dots011\dots1 \ \overline{b_{56}} \ \overline{b_{57}} \dots \overline{b_{63}}$$

【0068】サブケース1 ビット位置55について考える。ビット位置55は $\kappa_{56}=0$ に加算すべき1を2個有するので、 $\delta_{55}=0$ である。したがって、このサブケースでは、 $0 \leq i \leq 55$ である i について δ_i がすべて1ではない。

【0069】サブケース2 $\kappa_{56}=1$ であり、かつ $40 \leq i \leq 55$ である各ビット位置がその位置へのキャリーに加算すべき1を2個有するので、 $40 \leq i \leq 55$ であるすべての i について $\delta_i=1$ かつ $\kappa_i=1$ である。ビット位置39では、 $\kappa_{40}=1$ に0が2個加算される。したがって $\delta_{39}=1$ かつ $\kappa_{39}=0$ である。 $0 \leq i \leq 38$ である各ビット位置では、1とそのビット位置へのキャリー

28

【0063】64ビット・アドレス指定 Bが、Aが存在する256バイト・ブロックのすぐ上の256バイト・ブロック中にあるので、 $1 \leq j \leq 55$ であって、 $a_j=0$ かつ $b_j=1$ であり、 $j < p \leq 55$ である p について $a_p=1$ かつ $b_p=0$ となる位置 j が存在する。さらに、 $i < j$ の場合、 $a_i=b_i$ である。その結果、 $A-B$ は数式31で表すことができ、ホット1が暗黙のうちに強制的に位置63に入れられる。

* 【数31】

※ケース1は、Aがアドレス空間の高位256バイト・ブロック内にあり、Bが低位256バイト・ブロック内にあり、Aの最上位ビットがBの最上位ビットと等しいことを特徴とする。Aがアドレス空間の高位256バイト・ブロック内にあり、Bが低位256バイト・ブロック内にあるので、BはAより上にマップしなければならない。さらに、 $A(56:63)$ が $B(56:63)$ 未満である可能性があり、その場合は定理2により $\kappa_{56}=0$ (サブケース1)である。また、 $A(56:63)$ が $B(56:63)$ 以上である可能性もあり、その場合は定理1により $\kappa_{56}=1$ (サブケース2)である。3つのアドレス指定モードのすべてについてこの両方のサブケースを考慮しなければならない。

【0067】AR24ビット・モード このアドレス指定モードでは、BをAより上にマップするために、ビット39を強制的に論理1する。ビット39より上位のビット位置は等しいので、 $i \leq 38$ である i について $a_i=b_i$ である。その結果、 $A-B$ は数式32で表すことができ、ホット1が暗黙のうちに強制的にビット位置63に入れられる。

* 【数32】

とに0が加算される。 $\kappa_{39}=0$ なので、 $0 \leq i \leq 38$ であるすべての i について $\delta_i=1$ であり、したがって $0 \leq i \leq 55$ であるすべての i について $\delta_i=1$ である。

【0070】AR31ビット・モード このアドレス指定モードでは、BをAより上にマップするために、ビット32を強制的に論理1にする。ビット32より上位のビット位置は等しいので、 $i \leq 31$ である i について $a_i=b_i$ である。その結果、 $A-B$ は数式33で表すことができ、ホット1が暗黙のうちに強制的にビット位置63に入れられる。

* 【数33】

$$a_0 \ a_1 \ \dots \ a_{31} \ 01 \dots 1 \ a_{56} \ a_{57} \ \dots \ a_{63}$$

$$\overline{a_0} \ \overline{a_1} \ \dots \ \overline{a_{31}} \ 01 \dots 1 \ \overline{b_{56}} \ \overline{b_{57}} \ \dots \ \overline{b_{63}}$$

【0071】サブケース1 ビット位置55について考える。ビット位置55は、 $\kappa_{56}=0$ と加算すべき1を2個有するので、 $\delta_{55}=0$ である。したがってこのサブケースでは、 $0 \leq i \leq 55$ である*i*について δ_i はすべて1ではない。

【0072】サブケース2 $\kappa_{56}=1$ であり、かつ33 $\leq i \leq 55$ である各ビット位置がその位置へのキャリーと加算すべき1を2個有するので、 $33 \leq i \leq 55$ であるすべての*i*について $\delta_i=1$ かつ $\kappa_i=1$ である。ビット位置32では、 $\kappa_{33}=1$ に2個の0が加算される。したがって、 $\delta_{32}=1$ かつ $\kappa_{32}=0$ である。 $0 \leq i \leq 31$ である各ビット位置について、1とそのビット位置へのキャリーに0が加算される。 $\kappa_{32}=0$ であるので、 $0 \leq i \leq 31$ であるすべての*i*について $\delta_i=1$ であり、したがって $0 \leq i \leq 55$ であるすべての*i*について $\delta_i=1$ である。

*20

$$a_0 \ a_1 \ \dots \ a_{31} \ 00 \dots 011 \dots 1 \ a_{56} \ a_{57} \ \dots \ a_{63}$$

$$\overline{a_0} \ \overline{a_1} \ \dots \ \overline{a_{31}} \ 11 \dots 100 \dots 0 \ \overline{b_{56}} \ \overline{b_{57}} \ \dots \ \overline{b_{63}}$$

【0076】 $\kappa_{56}=1$ であり、かつ $0 \leq i \leq 55$ である各ビット位置がそのビット位置へのキャリーに加算すべき1を1個と0を1個有するので、 $0 \leq i \leq 55$ であるすべての*i*について $\delta_i=0$ かつ $\kappa_i=1$ である。したがって、 $0 \leq i \leq 55$ であるすべての*i*について $\delta_i \neq 1$ である。

*30

【数35】

$$a_0 \ a_1 \ \dots \ a_{31} \ 01 \dots 1 \ a_{56} \ a_{57} \ \dots \ a_{63}$$

$$\overline{a_0} \ \overline{a_1} \ \dots \ \overline{a_{31}} \ 10 \dots 0 \ \overline{b_{56}} \ \overline{b_{57}} \ \dots \ \overline{b_{63}}$$

【0078】 $\kappa_{56}=1$ であり、かつ $0 \leq i \leq 55$ である各ビット位置がそのビット位置へのキャリーに加算すべき1を1個と0を1個有するので、 $\delta_i=0$ かつ $\kappa_i=1$ である。したがって、 $0 \leq i \leq 55$ であるすべての*i*について $\delta_i \neq 1$ である。

【0079】64ビット・アドレス指定 64ビット・アドレス指定ではケース2は起こり得ない。したがって、これを扱う必要はない。

【0080】定理4 ケース3 ケース3は、BとAがアドレス空間の同じ256バイト・ブロック内にあり、★

$$a_0 \ a_1 \ \dots \ a_{31} \ 00 \dots 0 \ a_{40} \ a_{41} \ \dots \ a_{55} \ a_{56} \ a_{57} \ \dots \ a_{63}$$

$$\overline{a_0} \ \overline{a_1} \ \dots \ \overline{a_{31}} \ 11 \dots 1 \ \overline{a_{40}} \ \overline{a_{41}} \ \dots \ \overline{a_{55}} \ \overline{b_{56}} \ \overline{b_{57}} \ \dots \ \overline{b_{63}}$$

【0082】 $\kappa_{56}=0$ であり、かつ $0 \leq i \leq 55$ である各ビット位置がそのビット位置へのキャリーと加算すべ

*【0073】64ビット・アドレス指定 64ビット・アドレス指定ではケース1は起こり得ない。したがって、このケースの64ビット・アドレス指定モードではBをAより上にマップする必要はない。証明終り。

【0074】定理4 ケース2 ケース2は、BとAが共にアドレス空間の最高位256バイト・ブロック内にあり、 $A(56:63) \geq B(56:63)$ であることを特徴とする。したがって、定理1により $\kappa_{56}=1$ である。

【0075】AR24ビット・モード ビット40より上位のビット位置はBとAで等しいので、 $i \leq 39$ である*i*について $a_i=b_i$ である。その結果、A-Bは数式34で表すことができ、ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【数34】

*【0077】AR31ビット・モード ビット33より上位のビット位置はBとAで等しいので、 $i \leq 32$ である*i*について $a_i=b_i$ である。その結果、A-Bは数式35で表すことができる。ホット1が暗黙のうちに強制的にビット位置63に入れられる。

*30

【数35】

★ $B(56:63) > E(56:63) > A(56:63)$ であることを特徴とする。すなわち、 $A(56:63) < B(56:63)$ であるので、定理2により $\kappa_{56}=0$ である。256バイト・ブロックは任意である。

【0081】AR24ビット・モード ビット40より上位のビット位置は等しいので、 $i \leq 39$ である*i*について $a_i=b_i$ である。その結果、A-Bは数式36で表すことができる。ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【数36】

き1を1個と0を1個有するので、各ビット位置について $\delta_i=1$ かつ $\kappa_i=0$ である。したがって、 $0 \leq i \leq 55$

31

5であるすべての i について $\delta_i=1$ である。

【0083】AR31ビット・モード ビット33より上位のビット位置が等しいので、 $i \leq 32$ である i について $a_i=b_i$ である。その結果、 $A-B$ は数式37で表*

$$a_0 \ a_1 \ \dots \ a_{31} \ 0 \ a_{33} \dots a_{55} \ a_{56} \ a_{57} \ \dots \ a_{63}$$

$$\overline{a_0} \ \overline{a_1} \ \dots \ \overline{a_{31}} \ 1 \ \overline{a_{33}} \dots \overline{a_{55}} \ \overline{b_{56}} \ \overline{b_{57}} \ \dots \ \overline{b_{63}}$$

【0084】 $\kappa_{56}=0$ であり、かつ $0 \leq i \leq 55$ である各ビット位置がそのビット位置へのキャリーに加算すべき1を1個と0を1個有するので、各ビット位置について $\delta_i=1$ かつ $\kappa_i=0$ である。したがって、 $0 \leq i \leq 55$ であるすべての i について $\delta_i=1$ である。

【0085】64ビット・アドレス指定 64ビット・アドレス指定モードでは $A-B$ は数式38で表すことができる。ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【数38】

$$0 \ a_1 \ \dots \ a_{55} \ a_{56} \ a_{57} \ \dots \ a_{63}$$

$$1 \ \overline{a_1} \ \dots \ \overline{a_{55}} \ \overline{b_{56}} \ \overline{b_{57}} \ \dots \ \overline{b_{63}}$$

【0086】 $\kappa_{56}=0$ であり、かつ $0 \leq i \leq 55$ である各ビット位置がそのビット位置へのキャリーに加算すべ*

$$a_0 \ a_1 \ \dots \ a_{31} \ 0 \ 0 \dots 0 \ a_{40} \ a_{41} \dots a_{55} \ a_{56} \ a_{57} \ \dots \ a_{63}$$

$$\overline{a_0} \ \overline{a_1} \ \dots \ \overline{a_{31}} \ 1 \ 1 \dots 1 \ \overline{a_{40}} \ \overline{a_{41}} \dots \overline{a_{55}} \ \overline{b_{56}} \ \overline{b_{57}} \ \dots \ \overline{b_{63}}$$

【0090】 $\kappa_{56}=1$ であり、かつ $0 \leq i \leq 55$ である各ビット位置がそのビット位置へのキャリーに加算すべき0を1個と1を1個有するので、各ビット位置について $\delta_i=0$ かつ $\kappa_i=1$ である。したがって、 $0 \leq i \leq 55$ であるすべての i について $\delta_i \neq 1$ である。

【0091】AR31ビット・モード ビット33より★

$$a_0 \ a_1 \ \dots \ a_{31} \ 0 \ a_{33} \dots a_{55} \ a_{56} \ a_{57} \ \dots \ a_{63}$$

$$\overline{a_0} \ \overline{a_1} \ \dots \ \overline{a_{31}} \ 1 \ \overline{a_{33}} \dots \overline{a_{55}} \ \overline{b_{56}} \ \overline{b_{57}} \ \dots \ \overline{b_{63}}$$

【0092】 $\kappa_{56}=1$ であり、かつ $0 \leq i \leq 55$ である各ビット位置がそのビット位置へのキャリーに加算すべき1を1個と0を1個有するので、各ビット位置について $\delta_i=0$ かつ $\kappa_i=1$ である。したがって、 $0 \leq i \leq 55$ であるすべての i について $\delta_i \neq 1$ である。

【0093】64ビット・アドレス指定 64ビット・アドレス指定モードでは $A-B$ は数式41で表すことができる。ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【数41】

32

*すことができる。ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【数37】

※き1を1個と0を1個有するので、各ビット位置について $\delta_i=0$ かつ $\kappa_i=0$ である。したがって、 $0 \leq i \leq 55$ であるすべての i について $\delta_i=0$ である。

【0087】ケース3について要約すると、 $0 \leq i \leq 55$ であるすべての i について $\delta_i=1$ である。

【0088】定理48 ケース4 及び5 この2つのケースは、BとAがアドレス空間の同じ256バイト・ブロック内にあり、 $A(56:63) \geq B(56:63)$ であることを特徴とする。したがって、定理により $\kappa_{56}=1$ である。256バイト・ブロックは任意である。

【0089】AR24ビット・モード ビット40より上位のビット位置は等しいので、 $i \leq 39$ である i について $a_i=b_i$ である。その結果、 $A-B$ は次式のように表すことができる。ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【数39】

★上位のビット位置は等しいので、 $i \leq 32$ である i について $a_i=b_i$ である。その結果、 $A-B$ は数式40で表すことができる。ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【数40】

$$0 \ a_1 \ \dots \ a_{55} \ a_{56} \ a_{57} \ \dots \ a_{63}$$

$$1 \ \overline{a_1} \ \dots \ \overline{a_{55}} \ \overline{b_{56}} \ \overline{b_{57}} \ \dots \ \overline{b_{63}}$$

【0094】 $\kappa_{56}=1$ であり、かつ $0 \leq i \leq 55$ である各ビット位置がそのビット位置へのキャリーに加算すべき1を1個と0を1個有するので、各ビット位置について $\delta_i=0$ かつ $\kappa_i=1$ である。したがって、 $0 \leq i \leq 55$ であるすべての i について $\delta_i \neq 1$ である。

【0095】ケース4と5について要約すると、 $0 \leq i \leq 55$ であるすべての i について $\delta_i \neq 1$ である。

50 【0096】定理4 ケース6 及び11 この2つのケ

ースでは、Aのある256バイト・ブロックのすぐ上の256バイト・ブロックにBがある。さらに、A(56:63)がB(56:63)未満である可能性があり、その場合は定理2から $\kappa_{56}=0$ である(サブケース1)。またA(56:63)がB(56:63)以上である可能性もあり、その場合は定理1から $\kappa_{56}=1$ である(サブケース2)。3つのアドレス指定モードのそれぞれについて両方のサブケースを考慮しなければならない。

*

$$a_0 \ a_1 \ \dots \ a_{31} \ 00 \dots 0 \ a_{40} \ a_{41} \ \dots \ a_{j-1} \ 011 \dots 1 \ a_{56} \ a_{57} \ \dots \ a_{63}$$

$$\overline{a_0} \ \overline{a_1} \ \dots \ \overline{a_{31}} \ 11 \dots 1 \ \overline{a_{40}} \ \overline{a_{41}} \ \dots \ \overline{a_{j-1}} \ 011 \dots 1 \ \overline{b_{56}} \ \overline{b_{57}} \ \dots \ \overline{b_{63}}$$

【0098】サブケース1 ビット位置55について考える。この位置では、2個の1が0である κ_{56} に加算される。その結果 $\delta_{55}=0$ であり、したがって $0 \leq i \leq 55$ であるすべての i について $\delta_i=0$ である。

【0099】サブケース2 $\kappa_{56}=1$ であり、かつ $(j+1) \leq i \leq 55$ である各ビット位置がそのビット位置へのキャリーに加算すべき1を2個有するので、 $(j+1) \leq i \leq 55$ であるすべての i について $\delta_i=1$ かつ $\kappa_i=1$ である。ビット位置 j では、2個の0が $\kappa_{j+1}=1$ に加算され、したがって、 $\delta_j=1$ かつ $\kappa_j=0$ である。0と $j-1$ の間にあるすべてのビットで1個の1と1個の0がそのビット位置へのキャリーに加算される。*

$$a_0 \ a_1 \ \dots \ a_{31} \ 0 \ a_{33} \ \dots \ a_{j-1} \ 011 \dots 1 \ a_{56} \ a_{57} \ \dots \ a_{63}$$

$$\overline{a_0} \ \overline{a_1} \ \dots \ \overline{a_{31}} \ 1 \ \overline{a_{33}} \ \dots \ \overline{a_{j-1}} \ 011 \dots 1 \ \overline{b_{56}} \ \overline{b_{57}} \ \dots \ \overline{b_{63}}$$

【0101】サブケース1 ビット位置55について考える。2個の1が $\kappa_{56}=0$ に加算される。その結果、 $\delta_{55}=0$ であり、したがって $0 \leq i \leq 55$ であるすべての i について $\delta_i \neq 1$ である。

【0102】サブケース2 $\kappa_{56}=1$ であり、かつ $(j+1) \leq i \leq 55$ である各ビット位置がそのビット位置へのキャリーに加算すべき1を2個有するので、 $(j+1) \leq i \leq 55$ であるすべての i について $\delta_i=1$ かつ $\kappa_i=1$ である。ビット位置 j では、2個の0が $\kappa_{j+1}=1$ に加算され、したがって $\delta_j=1$ かつ $\kappa_j=0$ である。0と $j-1$ の間にある各ビット位置では、1個の0と1

40 個の1がそのビット位置へのキャリーに加算される。 $\kappa \star$

$$0 \ a_1 \ \dots \ a_{j-1} \ 011 \dots 1 \ a_{56} \ a_{57} \ \dots \ a_{63}$$

$$1 \ \overline{a_1} \ \dots \ \overline{a_{j-1}} \ 011 \dots 1 \ \overline{b_{56}} \ \overline{b_{57}} \ \dots \ \overline{b_{63}}$$

【0104】サブケース1 ビット位置55について考える。この位置では、2個の1が $\kappa_{56}=0$ に加算される。その結果、 $\delta_{55}=0$ であり、したがって $0 \leq i \leq 55$ であるすべての i について $\delta_i \neq 1$ である。

【0105】サブケース2 $\kappa_{56}=1$ であり、かつ $(j$

*【0097】AR24ビット・モード Aがある256バイト・ブロックのすぐ上の256バイト・ブロックにBがあるので、 $40 \leq j \leq 55$ であって、 $a_j=0$ かつ $b_j=1$ であり、 $j < p \leq 55$ である p について $a_p=1$ かつ $b_p=0$ である位置 j が存在する。さらに、 $i < j$ である i について $a_i=b_i$ である。その結果、 $A-B$ は次式のように表すことができる。ホット1が暗黙のうちにビット位置63に強制される。

* 【数42】

※ $\kappa_j=0$ であるので、 $0 \leq i \leq (j-1)$ であるすべての i について $\delta_i=1$ かつ $\kappa_i=0$ である。したがって、 $0 \leq i \leq 55$ である i について $\delta_i=1$ である。

【0100】AR31ビット・アドレス指定 Aがある256バイト・ブロックのすぐ上の256バイト・ブロックにBがあるので、 $33 \leq j \leq 55$ であって、 $a_j=0$ かつ $b_j=1$ であり、 $j < p \leq 55$ である p について $a_p=1$ かつ $b_p=0$ となる j が存在する。さらに、 $i < j$ である i について $a_i=b_i$ である。その結果、 $A-B$ は数式43で表すことができる。ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【数43】

$$a_0 \ a_1 \ \dots \ a_{31} \ 0 \ a_{33} \ \dots \ a_{j-1} \ 011 \dots 1 \ a_{56} \ a_{57} \ \dots \ a_{63}$$

$$\overline{a_0} \ \overline{a_1} \ \dots \ \overline{a_{31}} \ 1 \ \overline{a_{33}} \ \dots \ \overline{a_{j-1}} \ 011 \dots 1 \ \overline{b_{56}} \ \overline{b_{57}} \ \dots \ \overline{b_{63}}$$

30 $\star_j=0$ であるので、 $0 \leq i \leq (j-1)$ であるすべての i について $\delta_i=1$ かつ $\kappa_i=0$ である。したがって、 $0 \leq i \leq 55$ である i について $\delta_i=1$ である。

【0103】64ビット・アドレス指定 Aがある256バイト・ブロックのすぐ上の256バイト・ブロックにBがあるので、 $0 \leq j \leq 55$ であって、 $a_j=0$ かつ $b_j=1$ であり、 $j \leq p \leq 55$ である p について $a_p=1$ かつ $b_p=0$ となる j が存在する。さらに、 $i < j$ である i について $a_i=b_i$ である。その結果、 $A-B$ は数式44で表すことができる。ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【数44】

$$0 \ a_1 \ \dots \ a_{j-1} \ 011 \dots 1 \ a_{56} \ a_{57} \ \dots \ a_{63}$$

$$1 \ \overline{a_1} \ \dots \ \overline{a_{j-1}} \ 011 \dots 1 \ \overline{b_{56}} \ \overline{b_{57}} \ \dots \ \overline{b_{63}}$$

$+1) \leq i \leq 55$ である各ビット位置がそのビット位置へのキャリーと加算すべき1を2個有するので、 $(j+1) \leq i \leq 55$ であるすべての i について $\delta_i=1$ かつ $\kappa_i=1$ である。ビット位置 j では、2個の0が $\kappa_{j+1}=1$ に加算され、したがって、 $\delta_j=1$ かつ $\kappa_j=0$ であ

35

る。0と $j-1$ の間にある各ビット位置では、1個の1と1個の0がそのビット位置へのキャリーに加算される。 $\kappa_j=1$ であるので、 $0 \leq i \leq j-1$ であるすべての i について $\delta_i=1$ かつ $\kappa_i=0$ である。したがって、 $0 \leq i \leq 55$ である i について $\delta_i=1$ である。

【0106】ケース6及び11について要約すると、 $\kappa_{56}=0$ の場合、 $0 \leq i \leq 55$ であるすべての i について $\delta_i \neq 1$ である。 $\kappa_{56}=1$ の場合は、 $0 \leq i \leq 55$ であるすべての i について $\delta_i=1$ である。

【0107】定理4 ケース7 このケースでは、Bがある256バイト・ブロックのすぐ上の256バイト・ブロックにAがある。A(56:63)はB(56:63)未満である可能性があり、その場合は定理2から $\kappa_{56}=0$ である(サブケース1)。またA(56:63)*

$$a_0 \ a_1 \ \dots \ a_{31} \ 00 \dots 0 \ a_{40} \ a_{41} \ \dots \ a_{j-1} \ 100 \dots 0 \ a_{56} \ a_{57} \ \dots \ a_{63}$$

$$\overline{a_0} \ \overline{a_1} \ \dots \ \overline{a_{31}} \ 11 \dots 1 \ \overline{a_{40}} \ \overline{a_{41}} \ \dots \ \overline{a_{j-1}} \ 100 \dots 0 \ \overline{b_{56}} \ \overline{b_{57}} \ \dots \ \overline{b_{63}}$$

【0109】ビット位置 j について考える。この位置では、2個の1がそのビット位置へのキャリーに加算される。その結果、 $\kappa_{j+1}=0$ であろうと $\kappa_{j+1}=1$ であろうと、 $\kappa_j=1$ である。ビット位置 $j-1$ では、1個の1と1個のゼロが κ_j に加算される。したがって、 $\kappa_j=1$ であるので $\delta_{j-1}=0$ であり、したがって $0 \leq i \leq 55$ であるすべての i について $\delta_i \neq 1$ である。

【0110】31ビット・アドレス指定のアクセス・レジスタ・Bがある256バイト・ブロックのすぐ上の2 ※

$$a_0 \ a_1 \ \dots \ a_{31} \ 0 \ a_{33} \ \dots \ a_{j-1} \ 100 \dots 0 \ a_{56} \ a_{57} \ \dots \ a_{63}$$

$$\overline{a_0} \ \overline{a_1} \ \dots \ \overline{a_{31}} \ 1 \ \overline{a_{33}} \ \dots \ \overline{a_{j-1}} \ 100 \dots 0 \ \overline{b_{56}} \ \overline{b_{57}} \ \dots \ \overline{b_{63}}$$

【0111】ビット位置 j について考える。この位置では、2個の1がそのビット位置へのキャリーと加算される。その結果、 $\kappa_{j+1}=0$ であろうと $\kappa_{j+1}=1$ であろうと、 $\kappa_j=1$ である。ビット位置 $j-1$ では、1個の1と1個の0が κ_j に加算される。したがって、 $\kappa_j=1$ である $\delta_{j-1}=0$ であり、したがって $0 \leq i \leq 55$ であるすべての i について $\delta_i \neq 1$ である。

【0112】64ビット・アドレス指定 Bのある25 ★

$$0 \ a_1 \ \dots \ a_{j-1} \ 100 \dots 0 \ a_{56} \ a_{57} \ \dots \ a_{63}$$

$$1 \ \overline{a_1} \ \dots \ \overline{a_{j-1}} \ 100 \dots 0 \ \overline{b_{56}} \ \overline{b_{57}} \ \dots \ \overline{b_{63}}$$

【0113】ビット位置 j について考える。この位置では、2個の1がそのビット位置へのキャリーと加算される。その結果、 $\kappa_{j+1}=0$ であろうと $\kappa_{j+1}=1$ であろうと、 $\kappa_j=1$ である。ビット位置 $j-1$ では、1個の1と1個の0が κ_j に加算される。したがって、 $\kappa_j=1$ であるので $\delta_{j-1}=0$ であり、したがって $0 \leq i \leq 55$ であるすべての i について $\delta_i \neq 1$ である。

36

*がB(56:63)以上である可能性もあり、その場合は定理1から $\kappa_{56}=1$ である(サブケース2)。各アドレス指定モードについて両方のサブケースを同時に考慮することができる。

【0108】AR24ビット・モード Bがある256バイト・ブロックのすぐ上の256バイト・ブロックにAがあるので、 $40 \leq j \leq 55$ であって、 $a_j=1$ かつ $b_j=0$ であり、 $j < p \leq 55$ である p について $a_p=0$ かつ $b_p=1$ となる j が存在する。さらに、 $i < j$ である i について $a_i=b_i$ である。その結果、 $A-B$ は数式45で表すことができる。ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【数45】

※56バイト・ブロックにAがあるので、 $33 \leq j \leq 55$ であって、 $a_j=1$ かつ $b_j=0$ であり、 $j < p \leq 55$ である p について $a_p=0$ かつ $b_p=1$ となる位置 j が存在する。さらに、 $i < j$ である i について $a_i=b_i$ である。その結果、 $A-B$ は数式46で表すことができる。ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【数46】

★6バイト・ブロックのすぐ上の256バイト・ブロックにAがあるので、 $1 \leq j \leq 55$ であって、 $a_j=1$ かつ $b_j=0$ であり、 $j < p \leq 55$ である p について $a_p=0$ かつ $b_p=1$ となる位置 j が存在する。さらに、 $i < j$ である i について $a_i=b_i$ である。その結果、 $A-B$ は数式47で表すことができる。ホット1が暗黙のうちに強制的にビット位置63に入れられる。

【数47】

【0114】ケース7について要約すると、 $0 \leq i \leq 55$ であるすべての i について $\delta_i \neq 1$ である。

【0115】定理4 ケース8、9、10、12 これらのケースでは、AとBのある256バイト・ブロックが少なくとも1個のバイト・ブロックで分離されている。これらのケースでは、すべてのアドレス・モードを同じに扱うことができる。これらのケースでは、ビット

位置1と55の間に、 $i < j$ であるすべての i について $a_j \neq b_j$ かつ $a_i = b_i$ であるビット位置 j が少なくとも1つ存在する。

【0116】まず $a_j = \text{バー}b_j = 1$ のときのビット位置 j について考える。この条件のときは、 $\kappa_{j+1} = 0$ であろうと $\kappa_{j+1} = 1$ であろうと、 $\kappa_j = 1$ である。位置 $j-1$ では、1個の1と1個の0が κ_j に加算されて $\delta_{j-1} = 0$ となり、したがって $0 \leq i \leq 55$ であるすべての i について $\delta_i \neq 1$ である。

【0117】次に、 $a_j = \text{バー}b_j = 0$ かつ $\kappa_{j+1} = 0$ という条件について考える。 a_j 、 b_j バー及び κ_{j+1} の加算の結果は $\delta_j = 0$ であり、したがって $0 \leq i \leq 55$ であるすべての i について $\delta_i \neq 1$ である。

【0118】最後に、 $a_j = \text{バー}b_j = 0$ かつ $\kappa_{j+1} = 1$ の場合について考える。この場合は $\delta_j = 1$ かつ $\kappa_j = 0$ である。 $i < j$ である i について $\text{バー}b_i = \text{バー}a_i$ であり、かつ $\kappa_j = 0$ なので、 $0 \leq i < j$ であるすべての i について $\delta_i = 1$ かつ $\kappa_i = 0$ である。しかし $\kappa_{j+1} = 1$ の場合は、 $j+1$ の両方の加数が1であり、 $\kappa_{j+2} = 0$ である(サブケース1)か、または一方の加数が1、他方が0であり、 $\kappa_{j+2} = 1$ である(サブケース2)であるか、または両方の加数が1であり、 $\kappa_{j+2} = 1$ である(サブケース3)。サブケース1と2では、 $\delta_{j+1} = 0$ であり、したがって $0 \leq i \leq 55$ であるすべての i について $\delta_i \neq 1$ である。しかし、サブケース3では、 $\kappa_{j+2} = 1$ であるので $\delta_{j+1} = 1$ である。ビット位置 $j+2$ でも同じ3つのサブケースが生じる。したがって、 $\delta_{j+2} = 1$ の場合、 $j+2$ の両方の加数が1で $\kappa_{j+3} = 1$ でなければならず、あるいは $\delta_{j+2} = 0$ であり、したがって $0 \leq i \leq 55$ であるすべての i について $\delta_i \neq 1$ である。($j+1$) $\leq p \leq 55$ である p について同様の議論がビット毎に繰り返され、したがって ($j+1$) $\leq p \leq 55$ であるすべての p について $\delta_p = 1$ であり、したがって $a_p = \text{バー}b_p = 1$ である。したがって、 $a_p = 1$ かつ $b_p = 0$ であり、その結果、 $0 \leq i \leq 55$ であるすべての i について $\delta_i = 1$ ならばAが存在する256バイト・ブロックのすぐ上の256バイト・ブロックにBが存在しなければならなくなる。しかし、これはBとAが1つ以上の256バイト・ブロックで分離されているというこのケースに対する仮定と矛盾する。その結果、 $0 \leq i \leq 55$ であるすべての i について、 $\delta_i \neq 1$ である。

【0119】ケース8、9、10、12について要約すると、 $0 \leq i \leq 55$ であるすべての i について $\delta_i \neq 1$ である。

【0120】12のケースのそれぞれの考察からわかるように、記憶オペランド1及び2がMVC型のオーバーラップを含まない場合、 $\kappa_{56} = 1$ のときの非オーバーラップ・ケース1、6、11及び非オーバーラップ・ケース3を除き、 $0 \leq i \leq 55$ であるすべての i について δ

$i \neq 1$ である。証明終り。

【0121】定理5 $0 \leq i \leq 55$ であるすべての i について減算 $\text{SAR2} - \text{SAR1}$ の結果である δ_i が論理1であるなら、加算 ($\text{SAR2} - \text{SAR1}$) + L の間に生成されるビット位置56からビット位置55へのキャリーは、記憶オペランドがMVC型オーバーラップを含む場合、1である。

【0122】証明

if の証明

10 加算の累加性から、 $\text{SAR2}(55:63) - \text{SAR1}(55:63) + L(55:63)$ は、 $\Delta(55:63) + L(55:63)$ または $E(55:63) - B(55:63)$ のどちらかによって生成できる。その結果、数式48が成立する。

【数48】

$$S_{55} = S_{55}^{\dagger}$$

【0123】しかし、数式49が成立し、 $L_{55} = 0$ なので数式50が成立する。

20 【数49】

$$S_{55} = \delta_{55} \vee L_{55} \vee c_{56}$$

【数50】

$$S_{55} = \delta_{55} \vee c_{56}$$

【0124】数式51が成立するので、数式52が成立する。

【数51】

$$\delta_{55} = a_{55} \vee \text{バー}b_{55} \vee \kappa_{56}$$

【数52】

$$S_{55} = a_{55} \vee \text{バー}b_{55} \vee \kappa_{56} \vee c_{56}$$

【0125】同様に、数式53が成立するが、数式54が成立するので、数式55が成立する。

【数53】

$$S_{55}^{\dagger} = e_{55} \vee \text{バー}b_{55} \vee \lambda_{56}$$

【数54】

$$e_{55} = a_{55} \vee L_{55} \vee \gamma_{56} = a_{55} \vee \gamma_{56}$$

【数55】

$$S_{55}^{\dagger} = a_{55} \vee \gamma_{56} \vee \text{バー}b_{55} \vee \lambda_{56} = a_{55} \vee \text{バー}b_{55} \vee \gamma_{56} \vee \lambda_{56}$$

【0126】数式57が成立するので、数式57と58が成立する。

【数56】

$$S_{55} = S_{55}^{\dagger}$$

【数57】

$$a_{55} \vee \text{バー}b_{55} \vee \kappa_{56} \vee c_{56} = a_{55} \vee \text{バー}b_{55} \vee \gamma_{56} \vee \lambda_{56}$$

【数58】

$$\kappa_{56} \vee c_{56} = \gamma_{56} \vee \lambda_{56}$$

39

【 0 1 2 7 】ここで5 つのMVC 型オーバーラップ・ケースのそれぞれについて考察しなければならない。

【 0 1 2 8 】定理5 オーバーラップ・ケース1 及び5
ケース1 及び5 では、 $E(56:63) \geq B(56:63)$ である。定理1 により $\lambda_{56}=1$ である。さらに、 $A(56:63) > B(56:63)$ なので、定理1 により $\kappa_{56}=1$ である。最後に、 $A(56:63) + L(56:63)$ はアドレス空間の次の順次256 バイト・ブロックに入るので、キャリー γ_{56} も1 である。これを数式59 に代入すると、数式60 が得られる。

【 数59 】

$$\kappa_{56} \vee c_{56} = \gamma_{56} \vee \lambda_{56}$$

【 数60 】

$$1 \vee c_{56} = 1 \vee 1$$

$$\overline{c_{56}} = 0$$

$$c_{56} = 1$$

【 0 1 2 9 】定理5 オーバーラップ・ケース2 及び4
ケース2 及び4 では、 $A(56:63) < B(56:63)$ かつ $E(56:63) < B(56:63)$ である。定理2 により $\kappa_{56}=0$ かつ $\lambda_{56}=0$ である。ケース1 の場合と同様に $A(56:63) + L(56:63)$ は256 バイトの境界を横切るので、 $\gamma_{56}=1$ である。これらの恒等式から、数式61 が成立する。

【 数61 】

$$0 \vee c_{56} = 1 \vee 0$$

$$c_{56} = 1$$

【 0 1 3 0 】定理5 オーバーラップ・ケース3 ケース3 では、 $A(56:63) < B(56:63)$ かつ $E(56:63) \geq B(56:63)$ である。定理2 及び1 から $\kappa_{56}=0$ かつ $\lambda_{56}=1$ である。 $A(56:63) + L(56:63)$ は256 バイトの境界を横切らないので、 $\gamma_{56}=0$ である。これらの恒等式から、数式62 が成立する。

【 数62 】

$$0 \vee c_{56} = 0 \vee 1$$

$$c_{56} = 1$$

【 0 1 3 1 】only if の証明 only if を証明するには、 $0 \leq i \leq 55$ であるすべての i について $\delta_i=1$ であり、かつオペランドがMVC 型オーバーラップを含まないとき、 c_{56} が0 であることを示さなければならない。定理4 から、考察しなければならないケースは、非オーバーラップ・ケース3 と $\kappa_{56}=1$ のときのケース1、6、11 である。

40

【 0 1 3 2 】定理5 非オーバーラップ・ケース3 このケースでは、 $A(56:63) < B(56:63)$ かつ $E(56:63) < B(56:63)$ である。定理2 から $\kappa_{56}=0$ かつ $\lambda_{56}=0$ である。 $A(56:63) + L(56:63)$ は256 バイトの境界を横切らないので、 $\gamma_{56}=0$ である。したがって数式63 が成立する。

【 数63 】

$$0 \vee c_{56} = 0 \vee 0$$

$$c_{56} = 0$$

【 0 1 3 3 】定理5 非オーバーラップ・ケース1 及び6 これらのケースでは、 $\kappa_{56}=1$ の場合、 $0 \leq i \leq 55$ である。すべての i について $\delta_i=1$ である。したがって、 $\kappa_{56}=1$ である場合だけを考えればよい。さらに、 $E(56:63) < B(56:63)$ であるので、定理2 から $\lambda_{56}=0$ である。最後に、 $A(56:63) + L(56:63)$ は256 バイトの境界を横切るので、 $\gamma_{56}=1$ である。これらのキャリーを上記で導いたキャリー関係式に代入すると、数式64 が得られる。

【 数64 】

$$1 \vee c_{56} = 1 \vee 0$$

$$\overline{c_{56}} = 1$$

$$c_{56} = 0$$

【 0 1 3 4 】定理5 非オーバーラップ・ケース11 このケースでは、 $\kappa_{56}=1$ の場合、 $0 \leq i \leq 55$ である。すべての i について $\delta_i=1$ である。定理1 から、 $\kappa_{56}=1$ の場合、 $A(56:63) \geq B(56:63)$ である。しかし、 $E(56:63) \geq A(56:63)$ であるので、 $E(56:63) \geq B(56:63)$ である。この結果から、定理1 により $\lambda_{56}=1$ である。また $A(56:63) + L(56:63)$ は256 バイトの境界を横切らないので、 $\gamma_{56}=0$ である。これらのキャリーを上記で導いたキャリー関係式に代入すると数式65 が得られる。証明終り。

【 数65 】

$$1 \vee c_{56} = 0 \vee 1$$

$$\overline{c_{56}} = 1$$

$$c_{56} = 0$$

【 0 1 3 5 】定理6 SAR2 がアドレス空間の高位256 バイト・ブロックにあり、SAR1 が低位256 バイト・ブロックにあるとき、SAR1 がSAR2 より上にマップされ、演算 $(SAR2 - SAR1) + L$ の結果、ビット位置56 からビット位置55 へのキャリーが

41

生じ、 $0 \leq i \leq 55$ であるすべてのビット位置 i で結果がすべて0となるならば、記憶オペランド1及び2はMVC型オーバーラップを有する。

【0136】証明

if の証明

定理3により、2つのオペランドがMVC型オーバーラップを含み、SAR1が最低位256バイト・ブロックにありSAR2が最高位256バイト・ブロックにある場合にSAR1がSAR2のすぐ次の256バイト・ブロックにマップされるならば、SAR2-SAR1は0 $0 \leq i \leq 55$ である各ビット位置 i ですべて1を生成する。LはOL(0:7)とその左に連結された56個の0からなり、定理5から $c_{56}=1$ なので、 $0 \leq i \leq 55$ である各ビット i はそのビットへの1のキャリーに加算された1個の1と0からなる。その結果は0であり左隣のビットへのキャリーが生成され、したがって $0 \leq i \leq 55$ であるすべての i について結果は0となる。

【0137】only if の証明 記憶オペランド1及び2がMVC型オーバーラップを有しないと仮定する。さらに、 $0 \leq i \leq 55$ であるすべての i について $S_i=0$ であり、かつ $c_{56}=1$ であると仮定する。しかし、 $0 \leq i \leq 55$ であるすべての i について $S_i=0$ であるなら、 $c_{56}=1$ のとき、 $0 \leq i \leq 55$ であるすべての i について $S_i=1$ である。しかし、記憶オペランドがMVC型オーバーラップを有するならば、定理5から $0 \leq i \leq 55$ であるすべての i について $\delta_i=1$ であり、 $c_{56}=1$ である。したがって、記憶オペランドがMVC型オーバーラップを有しないという仮定は誤っている。証明終り。

【0138】定理6は、MVC型オーバーラップの検出に使用される条件を提供する。要約すると、MVC型オーバーラップでの検出は次のようにして行う。まず、SAR2が仮想アドレス空間の最高位256バイト・ブロックにある(SAR2HI)とき、SAR1が最低位256バイト・ブロックにある(SAR1LO)ことを検出しなければならない。このケースが生じる(REMAP1)場合、SAR1を最高位256バイト・ブロックのすぐ上の256バイト・ブロックにマップしなければならない。そうでない場合は、SAR1は再マップされない。このSAR1のマッピングでは、SAR1をSA*40

$$(SAR2 - SAR1) + L = (B_2 + D_2 - B_1 - D_1) + L$$

【0142】この議論では、SAR2とSAR1のどちらの計算も折り返さないものと仮定する。言い換えれば、24ビット・アドレス指定ではビット40から39へのキャリーが生じず、31ビット・アドレス指定ではビット33から32へのキャリーが生じず、64ビット・アドレス指定では先に論じたように折返しが発生し得ない。この折返しがないとの仮定により、当面の議論で

42

*R2から減算する。その結果に、56個の0とオペランド長の連結であるLを加算する。この加算の結果が $0 \leq i \leq 55$ であるすべての i についてすべて0を含み、ビット56から55へのキャリーが1であるならば、オペランドはMVC型オーバーラップを含む。次に、これらの条件をさらに展開して、MVC型オーバーラップのハードウェア予測に適した形にすることを追求する。

【0139】MVCオーバーラップを示す可能性のあるESA/390命令のアドレス計算は、基底に12ビットの変位を加算することからなる。24ビットのアドレス指定を用いるアクセス・レジスタ・モードでは、この加算は、24ビットの基底に8個のゼロを連結したものと連結したアクセス・レジスタに12ビットの変位を加算することからなる。31ビット・アドレス指定を用いるアクセス・レジスタ・モードでは、この加算は、31ビットの基底に1個のゼロを連結したものと連結したアクセス・レジスタに12ビットの変位を加算することからなる。64ビット・モードでは、その計算は、63ビットの基底に連結した1個の0に12ビットの変位を加算することからなる。以下では、12ビットの変位DP(0:11)の左に52個の0を連結した64ビット数を D_2 と D_1 で表すことにする。この形の記憶オペランド2に対する変位を D_2 で表し、記憶オペランド1に対する変位を D_1 で表す。 D_2 及び D_1 の下位12ビットをそれぞれ d_{2i} 及び d_{1i} で表す。ただし、 $52 \leq i \leq 63$ である。さらに、オペランド1及び2の基底をそれぞれ B_1 及び B_2 で表し、個々のビットを b_{1i} 及び b_{2i} で表す。ただし、 $0 \leq i \leq 63$ である。この形で表した基底は、上記のアクセス・レジスタ・モードの実際の基底とアクセス・レジスタの連結を含むことを理解されたい。前にSAR1を表すのに使用したBがここではアドレス計算の基底を表すことに留意されたい。

【0140】この表記法を使うと、SAR2とSAR1は数式66のように計算される。

$$\text{【数66】 } SAR2 = B_2 + D_2$$

$$SAR1 = B_1 + D_1$$

【0141】そうすると、(SAR2-SAR1)+Lは数式67のようになる。

【数67】

これらのビット位置へのキャリーを抑制するという複雑さを回避することができる。この複雑さについては後で考察する。

【0143】2の補数の算術を使用すると、(SAR2-SAR1)+Lの計算を数式68で表すことができる。

【数68】

$$(SAR2 - SAR1) + L = (B_2 + D_2 + B_1 + D_1 + 1 + 1) + L$$

43.

44

【 0 1 4 4 】 上式でバー B_i 及びバー D_i はそれぞれ B_i * 9 のような2 - 1 加算に還元できる。

及び D_i の補数を示す。括弧内の結果の計算は、数式6 * 【 数6 9 】

$$\begin{array}{cccccccccccccccc} b_{20} & b_{21} & \dots & b_{251} & b_{252} & b_{253} & b_{254} & b_{255} & b_{256} & \dots & b_{262} & b_{263} \\ 0 & 0 & \dots & 0 & d_{252} & d_{253} & d_{254} & d_{255} & d_{256} & \dots & d_{262} & d_{263} \end{array}$$

$$\overline{b_{10}} \ \overline{b_{11}} \ \dots \ \overline{b_{151}} \ \overline{b_{152}} \ \overline{b_{153}} \ \overline{b_{154}} \ \overline{b_{155}} \ \overline{b_{156}} \ \dots \ \overline{b_{162}} \ \overline{b_{163}}$$

$$e_0 \ e_1 \ \dots \ e_{51} \ e_{52} \ e_{53} \ e_{54} \ e_{55} \ e_{56} \ \dots \ e_{62} \ e_{63}$$

$$f_1 \ f_2 \ \dots \ f_{52} \ f_{53} \ f_{54} \ f_{55} \ f_{56} \ f_{57} \ \dots \ f_{63} \ 1$$

$$1 \quad 1 \quad \dots \ 1 \quad \overline{d_{152}} \ \overline{d_{153}} \ \overline{d_{154}} \ \overline{d_{155}} \ \overline{d_{156}} \ \dots \ \overline{d_{162}} \ \overline{d_{163}}$$

$$g_0 \ g_1 \ \dots \ g_{51} \ g_{52} \ g_{53} \ g_{54} \ g_{55} \ g_{56} \ \dots \ g_{62} \ g_{63}$$

$$h_1 \ h_2 \ \dots \ h_{52} \ h_{53} \ h_{54} \ h_{55} \ h_{56} \ h_{57} \ \dots \ h_{63} \ 1$$

【 0 1 4 5 】 個別ビット g_i を有する G と、 H が左に1ビットずつシフトされ、シフトによる空所が「ホット1」で埋められた、個別ビット h_i を有する H の2 - 1 20 加算の結果、 $B_2 + D_2 - B_1 - D_2$ となる。

【 0 1 4 6 】 この時点で、これまで使用してきた表記法を修正して、 $SAR_2 - SAR_1$ を求める際の上記の考察を含める必要がある。 G とシフトされた H の2 - 1 加算から、 $SAR_2 - SAR_1$ が得られるが、これを前に個別ビット δ_i を有する Δ で表したので、 G とシフトされた H の2 - 1 加算の結果をこの表記法で表すことにする。したがって、数式7 0 が成立する。

【 数7 0 】

$$g_0 \ \dots \ g_{55} \ g_{56} \ \dots \ g_{62} \ g_{63}$$

$$h_1 \ \dots \ h_{56} \ h_{57} \ \dots \ h_{63} \ 1$$

$$\delta_0 \ \dots \ \delta_{55} \ \delta_{56} \ \dots \ \delta_{62} \ \delta_{63}$$

【 0 1 4 7 】 同様に、 κ_i は、この2 - 1 加算ではビット i から $i - 1$ へのキャリーという意味になる。さらに、以前の表記法との整合性を保つため、 Δ と L の2 - 1 加算の結果を S で表し、その個別ビットを S_i で表すことにする。したがって、数式7 1 が成立する。

【 数7 1 】

$$\delta_0 \ \dots \ \delta_{55} \ \delta_{56} \ \dots \ \delta_{62} \ \delta_{63}$$

$$L_0 \ \dots \ L_{55} \ L_{56} \ \dots \ L_{62} \ L_{63}$$

$$S_0 \ \dots \ S_{55} \ S_{56} \ \dots \ S_{62} \ S_{63}$$

【 0 1 4 8 】 この2 - 1 加算でのビット位置 i から $i - 1$ へのキャリーを c_i で表すことにする。

【 0 1 4 9 】 $SAR_2 - SAR_1 + L$ の結果 S は、 G と 50

シフトされた H と L の $CSA_3 - 2$ 加算として計算することもでき、個別ビット σ_i を有する和と個別ビット ω_i を有するキャリーを生成する。個別ビット ω_i は、加算の前に左へ1ビット位置シフトされた ω と加算される。この2 - 1 加算の結果は $SAR_2 - SAR_1 + L$ を生じるが、これを個別ビット Σ_i を有する Σ で表して、2つの2 - 1 加算器を使った計算と区別する。したがって、数式7 2 のようになる。

【 数7 2 】

$$g_0 \ \dots \ g_{55} \ g_{56} \ \dots \ g_{62} \ g_{63}$$

$$h_1 \ \dots \ h_{56} \ h_{57} \ \dots \ h_{63} \ 1$$

$$L_0 \ \dots \ L_{55} \ L_{56} \ \dots \ L_{62} \ L_{63}$$

$$\sigma_0 \ \dots \ \sigma_{55} \ \sigma_{56} \ \dots \ \sigma_{62} \ \sigma_{63}$$

$$\omega_1 \ \dots \ \omega_{56} \ \omega_{57} \ \dots \ \omega_{63} \ 0$$

$$\Sigma_0 \ \dots \ \Sigma_{55} \ \Sigma_{56} \ \dots \ \Sigma_{62} \ \Sigma_{63}$$

【 0 1 5 0 】 δ_i とシフトされた ω_i の2 - 1 加算における i から $i - 1$ へのキャリーを ϕ_i で表すことにする。

【 0 1 5 1 】 定理6 から、 $c_{56} = 1$ かつ $0 \leq i \leq 55$ であるすべての i について $S_i = 0$ である場合に、MVC 型オーバーラップが存在する。 $0 \leq i \leq 55$ であるすべての i について $S_i = 0$ であるとの予測を $S_{=0}$ で表し、MVC 型オーバーラップを 0_{MVC} で表すと、数式7 3 のようになる。

【 数7 3 】 $0_{MVC} = c_{56} S_{=0}$

【 0 1 5 2 】 0_{MVC} が1 であるためには、 c_{56} が1 でなければならず、したがって $c_{56} = 1$ の場合についてのみ $S_{=0}$ を決定すればよい。 $c_{56} = 1$ のときの $S_{=0}$ を決定することを数式7 4 で表すことにする。次の定理が、数式7 5 を求める基礎となる

45

【数74】

$$S \neq 0$$

【数75】

$$S \neq 0$$

【0153】定理7 $c_{56}=1$ である場合、 $\kappa_{56}=1$ のときは $\phi_{56}=\omega_{56}=1$ 、 $\kappa_{56}=0$ のときは $\phi_{56}=1$ かつ $\omega_{56}=0$ または $\phi_{56}=0$ かつ $\omega_{56}=1$ である。

【0154】証明 結合法則により、3つの数の加算は加算の順序にかかわらず同じ結果を生じるので、数式76が成立する。

【数76】 $S_{55} = \Sigma_{55}$

【0155】しかし、数式77と78が成立し、したがって数式79が成立する。

【数77】

$$S_{55} = c_{56} \vee \delta_{55} \vee L_{55} = c_{56} \vee \delta_{55} \vee 0 = c_{56} \vee \delta_{55}$$

【数78】

$$\Sigma_{55} = \omega_{56} \vee \sigma_{55} \vee \phi_{56}$$

【数79】

$$c_{56} \vee \delta_{55} = \omega_{56} \vee \sigma_{55} \vee \phi_{56}$$

【0156】数式80と81から数式82が得られる。

【数80】

$$\delta_{55} = g_{55} \vee h_{56} \vee \kappa_{56}$$

【数81】

$$\sigma_{55} = g_{55} \vee h_{56} \vee L_{55} = g_{55} \vee h_{56} \vee 0 = g_{55} \vee h_{56}$$

【数82】

$$c_{56} \vee g_{55} \vee h_{56} \vee \kappa_{56} = \omega_{56} \vee g_{55} \vee h_{56} \vee \phi_{56}$$

$$c_{56} \vee \kappa_{56} = \omega_{56} \vee \phi_{56}$$

【0157】 $c_{56}=1$ である場合、数式83が成立する。

【数83】

$$\overline{\kappa_{56}} = \omega_{56} \vee \phi_{56}$$

【0158】 $\kappa_{56}=0$ である場合、数式84が成立する。

【数84】

$$1 = \omega_{56} \vee \phi_{56}$$

【0159】この恒等式が満たされるのは、 $\omega_{56}=1$ かつ $\phi_{56}=0$ または $\omega_{56}=0$ かつ $\phi_{56}=1$ の場合である。したがって、 $\kappa_{56}=0$ のとき、 $\omega_{56}=1$ かつ $\phi_{56}=0$ または $\omega_{56}=0$ かつ $\phi_{56}=1$ である。

【0160】 $\kappa_{56}=1$ である場合、数式85が成立する。

【数85】

46

$$0 = \omega_{56} \vee \phi_{56}$$

【0161】 ω_{56} と ϕ_{56} がこの式を満足するには、 $\omega_{56} = \phi_{56} = 1$ または $\omega_{56} = \phi_{56} = 1$ である。後は、 $\omega_{56} = \phi_{56} = 1$ であることを示せばよい。 κ_{56} の式は数式86で現せる。

【数86】 $\kappa_{56} = g_{56} h_{57} + g_{56} \kappa_{57} + h_{57} \kappa_{57}$

【0162】まず、 $\kappa_{57}=0$ であると仮定する。この場合、数式87が成立し、したがって $\kappa_{56}=1$ である場合、 $g_{56}=1$ かつ $h_{57}=1$ である。したがって、数式88が成立するので $\omega_{56}=1$ である。

【数87】 $\kappa_{56} = g_{56} h_{57}$ 【数88】 $\omega_{56} = g_{56} h_{57} + L_{56} h_{57} + L_{56} g_{56}$

【0163】キャリ一関係式に $\omega_{56}=1$ 、 $c_{56}=1$ 、 $\kappa_{56}=1$ を代入すると、数式89が得られる。

【数89】

$$1 \vee 1 = 1 \vee \phi_{56}$$

$$0 = \overline{\phi_{56}}$$

$$\phi_{56} = 1$$

【0164】したがって、 $\kappa_{57}=0$ 、 $\kappa_{56}=1$ 、 $c_{56}=1$ のとき、 $\omega_{56}=1$ かつ $\phi_{56}=1$ である。

【0165】次に $\kappa_{57}=1$ であると仮定すると、数式90が成立する。

【数90】 $\kappa_{56} = g_{56} + h_{57}$

【0166】したがって、 $\kappa_{56}=1$ である場合、 $g_{56}=1$ または $h_{57}=1$ である。さらに、数式91が成立する。

30 【数91】 $c_{56} = \delta_{56} L_{56} + \delta_{56} c_{57} + L_{56} c_{57}$

【0167】 $c_{57}=1$ であると仮定すると、数式92が成立し、 $c_{56}=1$ なので $\delta_{56}=1$ または $L_{56}=1$ である。まず $L_{56}=1$ であると仮定すると、数式93が成立し、 $g_{56}=1$ または $h_{57}=1$ なので、数式94が得られる。

【数92】 $c_{56} = \delta_{56} + L_{56}$ 【数93】 $\omega_{56} = g_{56} + h_{57}$ 【数94】 $\omega_{56}=1$

【0168】キャリ一関係式に代入すると、数式95が得られる。

【数95】

$$1 \vee 1 = 1 \vee \phi_{56}$$

$$0 = \overline{\phi_{56}}$$

$$\phi_{56} = 1$$

【0169】今度は $\delta_{56}=1$ であると仮定すると、数式96が成立し、かつ $\kappa_{57}=1$ であると仮定したので、数式97が成立する。

50

47

【数96】

$$\delta_{56} = g_{56} \forall h_{57} \forall \kappa_{57}$$

【数97】

$$\delta_{56} = \overline{g_{56} \forall h_{57}}$$

$$1 = \overline{g_{56} \forall h_{57}}$$

【0170】しかし、 $g_{56}=1$ または $h_{57}=1$ なので、 $g_{56}=1$ かつ $h_{57}=1$ となる。したがって、数式98が成立するので、 $\omega_{56}=1$ となる。キャリ一関係式に代入すると数式99が得られる。

【数98】 $\omega_{56} = g_{56} h_{57} + g_{56} L_{56} + h_{57} L_{56}$

【数99】

$$1 \vee 1 = 1 \vee \phi_{56}$$

$$0 = \overline{\phi_{56}}$$

$$\phi_{56} = 1$$

【0171】したがって、 $\kappa_{57}=1$ 、 $c_{57}=1$ 、 $\kappa_{56}=1$ 、 $c_{56}=1$ の場合、 $\omega_{56}=1$ かつ $\phi_{56}=1$ である。

【0172】最後に $c_{57}=0$ であると仮定すると、数式100が成立し、したがって $c_{56}=1$ である場合、 $\delta_{56}=1$ かつ $L_{56}=1$ である。数式101が成立するので、数式102が得られる。

【数100】 $c_{56} = \delta_{56} L_{56} + \delta_{56} c_{57} + L_{56} c_{57}$
 $c_{56} = \delta_{56} L_{56}$

【数101】 $\omega_{56} = g_{56} h_{57} + g_{56} L_{56} + h_{57} L_{56}$ 30

【数102】 $\omega_{56} = g_{56} + h_{57}$

【0173】しかし、仮定したように $\kappa_{57}=1$ の場合は、 $g_{56}=1$ または $h_{57}=1$ であり、したがって $\omega_{56}=1$ となる。再度キャリ一関係式に代入すると、数式103が得られる。

【数103】

$$1 \vee 1 = 1 \vee \phi_{56}$$

$$0 = \overline{\phi_{56}}$$

$$\phi_{56} = 1$$

【0174】したがって、 $\kappa_{57}=1$ 、 $c_{57}=0$ 、 $\kappa_{56}=1$ 、 $c_{56}=1$ の場合、 $\omega_{56}=1$ かつ $\phi_{56}=1$ であり、したがって $c_{56}=1$ かつ $\kappa_{56}=1$ である場合、 $\omega_{56} = \phi_{56} = 1$ となる。証明終り。

【0175】次に $\Sigma(0:55)$ の計算について考える。 Σ は σ と左に1ビット位置シフトされた ω の2-1加算によって算出される。これは数式104で表される。

48

【数104】

$$\sigma_0 \sigma_1 \dots \sigma_{54} \sigma_{55}$$

$$\omega_1 \omega_2 \dots \omega_{55} \omega_{56}$$

$$\Sigma_0 \Sigma_1 \dots \Sigma_{54} \Sigma_{55}$$

【0176】 $c_{56}=1$ のとき、 $0 \leq i \leq 55$ である i についてすべての Σ_i が0であることを決定しなければならない。この決定を行うには、定理7を用いて Σ の計算の表現を修正する。定理7から、 $c_{56}=1$ のときは、 $\kappa_{56}=1$ のとき ϕ_{56} と ω_{56} が共に1でなければならない、 $\kappa_{56}=0$ のとき ϕ_{56} と ω_{56} のうち一方だけが1となり得る。したがって、この加算は数式105のような3-1加算として表すことができる。

【数105】

$$\sigma_0 \sigma_1 \dots \sigma_{54} \sigma_{55}$$

$$\omega_1 \omega_2 \dots \omega_{55} \omega_{56}$$

$$0 \quad 0 \dots 0 \quad 1$$

$$\Sigma_0 \Sigma_1 \dots \Sigma_{54} \Sigma_{55}$$

【0177】加算をこの形で表すことの利点は、下記の諸定理から出てくる。提示を簡単にするため、 $0 \leq i \leq 55$ であるすべての i について $\sigma_i = P_{xi}$ で表し、 κ_{56} と連結された ω_i ($1 \leq i \leq 55$) を P_{yi} ($1 \leq i \leq 56$) で表すことになる。そうすると上記の式は数式106のようになる。

【数106】

$$P_{x0} P_{x1} \dots P_{x54} P_{x55}$$

$$P_{y1} P_{y2} \dots P_{y55} P_{y56}$$

$$0 \quad 0 \dots 0 \quad 1$$

$$\Sigma_0 \Sigma_1 \dots \Sigma_{54} \Sigma_{55}$$

【0178】定理8 $\Sigma_i=0$ である場合、 P_x と P_y と1の加算によって生成される位置 i でのキャリ一 v_i は、 $0 \leq i \leq 55$ であるすべての i について1に等しい。

【0179】証明 定理8は帰納法で証明できる。

40 【0180】帰納の基底 ビット位置55における和の式は数式107のようになる。

【数107】

$$\Sigma_{55} = P_{x55} \vee P_{y55} \vee 1$$

$$= \overline{P_{x55} \vee P_{y55}}$$

【0181】 $\Sigma_{55}=0$ であると仮定すると数式108が成立し、したがって数式109が成立する。

【数108】

$$\overline{P_{x55} \vee P_{y55}} = 0,$$

【数109】

$$P_{x55} \vee P_{y56} = 1.$$

【0182】ビット位置55で生成されるキャリー v_{55} は数式110で表される。

【数110】

$$v_{55} = P_{x55} P_{y56} + P_{x55}(1) + P_{y56}(1)$$

【0183】上式で+は論理ORを表す。この式は数式111に還元される。

【数111】 $v_{55} = P_{x55} + P_{y56}$

【0184】 $\Sigma_{55} = 0$ の場合、数式112が成立し、かつ P_{x55} と P_{y56} のどちらか一方が1であるが、両方が1ではない場合は排他的ORは1となるので、数式113が成立し、したがって数式114が成立する。

【数112】

$$P_{x55} \vee P_{y56} = 1,$$

【数113】 $P_{x55} + P_{y56} = 1$ 【数114】 $v_{55} = 1$

【0185】したがって、ビット55での加算の和 Σ_{55} が0に等しい場合、ビット55からビット54へのキャリーは1であることが保証される。

【0186】帰納のステップ $\Sigma_i = 0$ かつ $v_{i+1} = 1$ であると仮定する。位置 i での和 Σ_i は数式115で表されるので、数式116または117が成立する。

【数115】

$$\Sigma_i = P_{xi} \vee P_{yi+1} \vee v_{i+1},$$

【数116】

$$P_{xi} \vee P_{yi+1} \vee 1 = 0,$$

【数117】

$$\overline{P_{xi} \vee P_{yi+1}} = 0$$

$$P_{xi} \vee P_{yi+1} = 1.$$

【0187】位置 i でのキャリーは数式118で表され *

$$v_i = P_{xi} P_{yi+1} + P_{xi} v_{i+1} + P_{yi+1} v_{i+1}$$

$$= P_{xi} P_{yi+1} + P_{xi} v_{i+1} (P_{yi+1} + \overline{P_{yi+1}}) P_{yi+1} v_{i+1} (P_{xi} + \overline{P_{xi}})$$

$$= P_{xi} P_{yi+1} + P_{xi} \overline{P_{yi+1}} v_{i+1} + P_{yi+1} \overline{P_{xi}} v_{i+1}$$

$$= P_{xi} P_{yi+1} + (P_{xi} \vee P_{yi+1}) v_{i+1}$$

【0194】仮定により、数式126が成立するので、数式127が成立する。

【数126】

$$P_{xi} \vee P_{yi+1} = 1,$$

【数127】 $v_i = v_{i+1}$

【0195】上式がすべての i について成立しなければならず、かつキャリー・インが1なので、数式128が成立する。

【数128】 $v_i = v_{i+1} = \dots = 1$ 【0196】 $\Sigma_i = 1$ とする。ただし Σ_i は数式129で

*る。

【数118】

$$v_i = P_{xi} P_{yi+1} + P_{xi} v_{i+1} + P_{yi+1} v_{i+1}$$

【0188】 $v_{i+1} = 1$ なので、数式119が成立する。

【数119】 $v_i = P_{xi} P_{yi+1} + P_{xi} + P_{yi+1}$

$$= P_{xi} + P_{yi+1}$$

【0189】しかし、上記から数式120が成立し、したがって数式121及び数式122が成立する。

【数120】

$$P_{xi} \vee P_{yi+1} = 1,$$

【数121】 $P_{xi} + P_{yi+1} = 1$ 【数122】 $v_i = 1$

【0190】上記のステップ1と2から、数学的帰納により $0 \leq i \leq 55$ である i について $v_i = 1$ となる。証明終り。

【0191】定理9 $0 \leq i \leq 55$ である i について数式123が成立するならば和 Σ は0である。

【数123】

$$P_{xi} \vee P_{yi+1} = 1$$

【0192】証明

if の証明

$\Sigma = 0$ ならば、任意の i について定理8から $v_{i+1} = 1$ であり、したがって $\Sigma_i = 0$ なので数式124が成立する。

【数124】

$$P_{xi} \vee P_{yi+1} = 1.$$

30 【0193】only if の証明

v_i で生成されるキャリーについて考える。このキャリーは数式125で表される。

【数125】

表される。

【数129】

$$\Sigma_i = P_{xi} \vee P_{yi+1} \vee v_{i+1}.$$

【0197】ただし、数式130が成立し、したがって数式131が成立する。

【数130】

$$P_{xi} \vee P_{yi+1} = 1$$

【数131】 $v_{i+1} = 0$ 【0198】したがって、 $\Sigma_i = 1$ であると仮定すると

51

矛盾が生じ、したがってこの仮定は誤っている。したがって Σ_i は 0 でなければならない。証明終り。

【 0 1 9 9 】 数式 1 3 2 が 0 となるための条件は、定理 8 と 9 から導かれる。この条件とは、 $0 \leq i \leq 55$ であるすべての i について数式 1 3 3 が成立することである。 P_{xi} と P_{yi+1} に適当な値を代入すると数式 1 3 4 が得られる。

【 数 1 3 2 】

$$S_{\perp 0}$$

【 数 1 3 3 】

$$P_{xi} \vee P_{yi+1} = 1$$

【 数 1 3 4 】

$$S_{\perp 0}^{\dagger} = (\sigma_0 \vee \omega_1) (\sigma_1 \vee \omega_2) \dots (\sigma_{54} \vee \omega_{55}) (\sigma_{55} \vee \kappa_{56})$$

【 0 2 0 0 】 数式 1 3 5 が決定されると、既に表示したように、数式 1 3 6 から 0 MVC を求めることができる。

【 数 1 3 5 】

$$S_{\perp 0}$$

【 数 1 3 6 】

$$O_{MVC} = c_{56} S_{\perp 0}$$

【 0 2 0 1 】 MVC 型オーバーラップを予測するための制限のある装置が開発されたばかりである。この装置は、第 1 及び第 2 のオペランド・アドレスの計算中にラップが起こらないと仮定されているので、一般的ではない。さらに、第 1 オペランド・アドレスが有効アドレス空間の最低位 2 5 6 バイト・ブロックを指すとき、第 2 オペランド・アドレスが最高位 2 5 6 バイト・ブロックを指さないと仮定されている。したがって、第 1 オペラ 30

ンド・アドレスを第 2 オペランド・アドレスより上に再マップする必要はなかった。一般の場合に MVC 型オーバーラップを予測するには、こうした状況を考慮して、この以前の予測方式をオーバーラップが正しく予測できるように補正しなければならない。

【 0 2 0 2 】 オーバーラップが正しく予測できるようにするために行わなければならない補正を表 3 に示す。この表に列挙した補正は 3 つのアドレス指定モードのすべてに適用される。表中で REMAP 1 は、SAR 2 がアドレス空間の最高位 2 5 6 バイト・ブロックにあり、SAR 1 が最低位 2 5 6 バイト・ブロックにあるときに発生する。これが発生するときは、SAR 2 が存在する 2 5 6 バイト・ブロックのすぐ上の 2 5 6 バイト・ブロックに SAR 1 を再マップしなければならない。これは、アドレス空間を指定するビットの左 MSB に隣接するビット位置を強制的に 1 にすることによって実施できる。このビット位置は、2 4 ビット・アドレス指定の場合はビット 3 9、3 1 ビット・アドレス指定の場合はビット 3 2 である。6 4 ビット・アドレス指定の場合は、前述の理由から再マップは不要である。SAR 1 及び SAR

52

2 のアドレス計算中に折返しが発生したことを示すのにそれぞれ WRAP 1 及び WRAP 2 を使用する。その際に基底が変位に加算される。WRAP 1 及び WRAP 2 の条件は、基底が変位と加算されるとき、アドレス空間を指定するビットの MSB からキャリーが生じることを検出することによって識別できる。したがって、2 4 ビット及び 3 1 ビットの場合に折返しが発生したことを判定するために、それぞれビット 4 0 及び 3 3 からのキャリーを検出しなければならない。6 4 ビット・アドレス指定での $2^{64}-1$ のアドレス空間及び 6 4 ビット・アドレスの MSB は 0 でなければならないので、6 4 ビット・アドレス指定モードでは折返しは生じ得ない。しかし 2 4 ビットまたは 3 1 ビット・アドレス指定モードで折返しが発生するときは、オーバーラップが正しく予測できるように、アドレス空間の MSB から生じるキャリーを抑制しなければならない。

【 0 2 0 3 】 上記の議論から、表 3 に列挙した補正を適用しなければならないのは、2 4 ビット及び 3 1 ビット・アドレス指定モードのときだけである。この 2 つのモードで必要な補正の違いは、キャリーを抑制または強制すべきビット位置だけである。このため、以下の議論では 2 4 ビット・アドレス指定の場合の補正の適用のみについて述べる。3 1 ビット・アドレス指定の場合の結果は、展開なしに提示されるであろう。前述のように、2 4 ビット・アドレス指定で強制または抑制しなければならないキャリーは、B と D の加算中に生成されるビット位置 4 0 からビット位置 3 9 へのキャリーである。このキャリーは、加算 B + D にその値を加算することによって強制できる。このキャリーの値は数式 1 3 7 で与えられ、ビット位置 3 9 に 1 が生じる。

【 数 1 3 7 】 $00 \dots 010 \dots 00$

【 0 2 0 4 】 同様に、キャリー値の 2 の補数を加えることによってキャリーが抑制できる。上式のキャリーの値を使うと、2 の補数は数式 1 3 8 のようになる。

【 数 1 3 8 】

$$11 \dots 101 \dots 11$$

$$00 \dots 000 \dots 01$$

$$11 \dots 110 \dots 00$$

【 0 2 0 5 】 したがって、B + D に $(11 \dots 110 \dots 00)$ を加えることによって、キャリーが抑制できる。しかし、MVC 型オーバーラップを予測するには、数式 1 3 9 の算術演算から $0 \leq i \leq 55$ であるビット位置についてすべて 0 を検出しなければならない。

【 数 1 3 9 】 $B_2 + D_2 - (B_1 + D_1) + L$

【 0 2 0 6 】 第 1 オペランドのアドレス計算によるキャリーを強制するには、キャリーの値を $B_1 + D_1$ に加算しなければならない。しかし、上式では $B_1 + D_1$ が減算さ

れるので、第1 オペランド・アドレスの計算のためにキャリーを強制するには、キャリー値の2の補数をMVC型オーバーラップでの計算に加算しなければならない。同様に、第1 オペランドのアドレス計算からのキャリーを抑制するには、上式にキャリー値を加算しなければならない。

【0207】この議論から、表3のケース1ないし4で適用すべき補正は明らかになるであろう。第1のケースでは補正は不要である。これは、 $B_2 + D_2$ 及び $B_1 + D_1$ の結果に0を加算することによって実施できる。その正味の効果は、 $B_2 + D_2 - (B_1 + D_1)$ に0を加算することである。第1オペランドの計算中に折返しが発生する第2のケースでは、 $B_1 + D_1$ の間に発生するキャリーを抑制しなければならない。したがって、このキャリーを抑制するには、上記の理由から、 $B_2 + D_2 - (B_1 + D_1)$ にキャリー値を加算しなければならない。ケース3では第2オペランドのアドレス計算中に折返しが発生する。この場合、このアドレス計算によるキャリーは、 $B_2 + D_2 - (B_1 + D_1)$ にキャリー値の2の補数を加算することによって抑制される。最後に第4のケースでは、両方のオペランドのアドレス計算による折返しを抑制しなければならない。しかし、これらのキャリーを抑制するために加算しなければならない値は加法逆数であり、したがって正味の結果として、この補正のために0の値を加算しなければならない。以上の結果を表4の最初の4行にまとめてある。

【0208】表3の残り4つのケースでは、アドレス空間の最高位256バイト・ブロックのすぐ上にSAR1を再マップする必要がある。前述のように、これは、SAR1を生成するためのアドレス計算の結果の適当なビット位置に強制的に1個の1を入れることによって実施される。これは、そのビット位置にキャリーを強制することと同じことである。したがって、どちらかのオペランド・アドレスの計算中に折返しが起こらないのに再マップが必要なケース5の場合は、キャリー値を $B_1 + D_1$ に加算しなければならない。したがって $B_2 + D_2 - (B_1 + D_1)$ にキャリー値の2の補数を加算しなければならない。ケース6の場合は、第1オペランドの計算で折返しが発生するのに、再マップが必要である。すなわち、 $B_1 + D_1$ がキャリーを発生し、それによってSAR1がアドレス空間の最高位256バイト・ブロックのすぐ上に自動的に再マップされる。 $B_2 + D_2 - (B_1 + D_1)$ に0を加算することによって実施できる補正は必要でない。24ビットの基底に12ビットの変位を加算しても、折返しは発生し得ず、24ビット・アドレス空間の最高位256バイト・ブロックにあるアドレスが発生し

得ないので、ケース7及び8は発生し得ない。したがって、これらのケースの対応する項目は表4に含まれていない。31ビット・アドレス指定での同じ分析の結果を表5に示す。表4及び5からわかるように、すべてのアドレス指定モードのすべてのケースの考察から、独特な5つの補正が生じる。これら独特な補正を表6にまとめて示す。これら独特な補正と表4及び5の各ケースの間のマップを表7に示す。

【0209】定理3ないし6に含まれる仮定と矛盾しないSAR2 - SAR1の値を生成するには、 $B_2 + D_2 - (B_1 + D_1)$ の計算に表6の補正を適用しなければならない。したがって、 $B_2 + D_2 - (B_1 + D_1)$ にオペランド記憶域長を加算する前に、補正値を適用しなければならない。これは、補正値を加算し、加算の結果がSAR2 - SAR1となる2つのエンティティを得るために3 - 2 CSA段がもう1つ必要なことを示唆する。これらの定理で要求されるように折返しを除去され、再マップが強制される。次にCSAの出力を上記のG及びシフトされたHと同様に扱ってオーバーラップを検出することができる。

【0210】しかし、次のようにしてさらに高い並列性を得ることができる。まず、3 - 2加算器を使って $B_2 + \text{バー}B_1 + \text{バー}D_1$ を和とキャリーに還元することができる。補正が行われない場合は、第2の3 - 2 CSAによってこの和とキャリーを D_2 と加算し、前述のように使用されるG及びシフトされたHを生成することができる。しかし、補正が必要な場合は、補正値を D_2 と連結することによって加算が実施できるので、 D_2 と補正値の2 - 1加算が直ちに生成できる。たとえば、表6の補正3で、補正値と D_2 の2 - 1加算は数式140のようになる。

【数140】

$$\begin{array}{cccccccc} 0 & 0 & \dots & 0 & 0 & 0 & \dots & d_{252} d_{253} \dots d_{263} \\ 1 & 1 & \dots & 1 & 1 & 0 & \dots & 0 \quad 0 \quad \dots \quad 0 \end{array}$$

$$1 \quad 1 \quad \dots \quad 1 \quad 1 \quad 0 \quad \dots \quad d_{252} d_{253} \dots d_{263}$$

【0211】次にこの値を $B_2 + \text{バー}B_1 + \text{バー}D_1$ の3 - 2加算からの和及びキャリーに加算すると、補正されたGとHを生成することができ、これは制限付きのケースでのオーバーラップ検出の場合と同様に扱うことができる。したがって数式141が得られる。

【数141】

$\begin{matrix} 55 & & & & & & & & & 56 \\ b_{20} & b_{21} & \dots & b_{251} & b_{252} & b_{253} & b_{254} & b_{255} & b_{256} & \dots & b_{262} & b_{263} \end{matrix}$									
$\begin{matrix} b_{10} & b_{11} & \dots & b_{151} & b_{152} & b_{153} & b_{154} & b_{155} & b_{156} & \dots & b_{162} & b_{163} \end{matrix}$									
$\begin{matrix} 1 & 1 & \dots & 1 & d_{152} & d_{153} & d_{154} & d_{155} & d_{156} & \dots & d_{162} & d_{163} \end{matrix}$									
$\begin{matrix} e_0 & e_1 & \dots & e_{51} & e_{52} & e_{53} & e_{54} & e_{55} & e_{56} & \dots & e_{62} & e_{63} \\ f_1 & f_2 & \dots & f_{52} & f_{53} & f_{54} & f_{55} & f_{56} & f_{57} & \dots & f_{63} & 1 \\ 1 & 1 & \dots & 0 & d_{252} & d_{253} & d_{254} & d_{255} & d_{256} & \dots & d_{262} & d_{263} \end{matrix}$									
$\begin{matrix} g_0 & g_1 & \dots & g_{51} & g_{52} & g_{53} & g_{54} & g_{55} & g_{56} & \dots & g_{62} & g_{63} \\ h_1 & h_2 & \dots & h_{52} & h_{53} & h_{54} & h_{55} & h_{56} & h_{57} & \dots & h_{63} & 1 \end{matrix}$									

【 0 2 1 2 】 最高の並列性を引き出すには、2 つの3 - 2 CSA のシーケンスによって $B_2 - B_1 - D_1 + D_2$ の計算に表6 に示したすべての潜在的補正を適用して、すべての潜在的G 及びH を生成することができる。これらを20 加算すると所望の $SAR_2 - SAR_1$ の値が得られる。これらはそれぞれ前に示したように使用され、そのケー

スが決定された後に適切な標識が使用のために選ばれる。ケースの決定は、表7 に従ってアドレス指定モード、WRAP1、WRAP2、REMAP1 の適切な番号により、これらの計算と平行して行われる。

【 表3 】

MVC型オペランドのオーバーラップを予測するために有効アドレス空間のMSVからのキャリーを強制または抑制するための要件				
ケース	REMAP1	VRAP2	VRAP1	処置
ケース1	なし	なし	なし	補正処置なし
ケース2	なし	なし	あり	$B_1 + D_1$ に対するMSBからのキャリーを抑制する
ケース3	なし	あり	なし	$B_2 + D_2$ に対するMSBからのキャリーを抑制する
ケース4	なし	あり	あり	$B_2 + D_2$ に対するMSB及び $B_1 + D_1$ からのキャリーを抑制する
ケース5	あり	なし	なし	$B_1 + D_1$ に対するMSBからのキャリーを強制する
ケース6	あり	なし	あり	$B_1 + D_1$ に対するMSBからのキャリーを許容する
ケース7	あり	あり	なし	不能
ケース8	あり	あり	あり	不能

24ビット・アドレス指定モードでのMVC型オーバーラップの検出用の補正

[illegible]

31ビット・アドレス指定モードでのMVC型オーバーラップの検出用の補正

[illegible]

【表6】

[illegible]

【表7】

31ビット・アドレス指定モードでのMVC型オーバーラップの提出用の補正	
補 正	マッピング
COR1	すべてのアドレス指定モード ケース1, 4, 6
COR2	AR24ビットモード ケース2
COR3	AR24ビットモード ケース3及び5
COR4	AR31ビットモード ケース 2
COR5	AR31ビットモード ケース3及び5

【 0 2 1 3 】 好ましい実施例

ここで本発明の詳細な説明に移ると、図4には、MVC型の破壊的な記憶オペランド・オーバーラップを予測するのに使用されるハードウェアの概要が示してある。レジスタ l_h 、 r_h 、 l_a 、 r_a はそれぞれSAR2及びSAR1を算出するための基底及び変位の値であるB2、D2、B1、D1を格納する。これらのレジスタは、折返し条件WRAP2とWRAP1を予測し、かつ図5及び図6に示すようにSAR2がアドレス空間の高位256バイト・ブロックにあるかどうか(SAR2H1)またはSAR1がアドレス空間の低位256バイト

40 ・ ブロックにあるかどうかを判定するためのブロックに
出力を供給する。SAR2HI と SAR2LO は、図4
に示すようにANDされて、再マップが必要なことを示
すREMAP1（図4には記載せず）を生成する。RE
MAP1、WRAP1、WRAP2 は、表6 に従ってア
ドレス指定モードと共に復号されて、オーバーラップを
判定するためにどのオーバーラップ補正（以下の式で
は、OMVCCOR1、OMVCCOR2、OMVCCOR3 で示す）を使用
すべきかを示すCOR（1：5）を生成する（図4 のR
EMAP1、REMAP2、WRAP1、WRAP2 復
50 号ブロック）。COR（1：5）はオーバーラップ予測

ブロックに送られ、そこで適当なオーバーラップ標識の実際の選択が行われる。オーバーラップ予測ブロックは、前述の諸アルゴリズムを実施したものである。このブロックの詳細は、図5 及び6 に示す。

【 0 2 1 4 】図5 及び6 では、キャリー・セーブ加算器 CSA1 を使って、 $l h + l a _N + r a _N (B_2 + \text{バー}B_1 + \text{バー}D_1)$ の3 - 2 加算を実行する。CSA1 によって生成される和E とキャリーF は、F が左に1 ビット 位置ずつシフトされ、右端のビット 位置に「 ホット 1 」が付加されて、2 つの3 - 2 加算器 CSA2 と CSA3 に供給される。CSA2 にはE とF の他に $r h (D_2)$ も供給される。したがって、CSA2 の結果は和G とキャリーH であり、これらは加算されると、H が左に1 ビット 位置ずつシフトされ、右端のビット 位置に「 ホット 1 」が供給されて、 $B_2 + D_2 - (B_1 + D_1)$ を生成することになる。その間にE とF の左端の4 0 ビット 位置が、CSA3 に送られて、4 0 個の1 と加算される。これによって、折返し条件及び再マップ条件の場合に $B_2 + D_2 - (B_1 + D_1)$ の計算を補正する際に使用される値が生成される (図5 及び6 及び本発明の挙動を指定する以下の諸式の ϵ と δ) 。CSA2 からのG とH は、2 - 1 加算器によって加算されるのではなく、CSA7 に供給され、そこでこれらの値にL が加えられて、和 σ とキャリー ω を生成する。それに並列に、 $G (56 : 63)$ と $H (57 : 63) || 0$ がキャリー生成器に供給されて、標準のキャリー先読み技法によって κ_{56} を生成する。その間にCSA4 がCSA2 の和の選択された組合せをCSA3 からのキャリーと加算して、和 ν とキャリーN を生成し、CSA5 がCSA3 の和をCSA2 のキャリーと加算して、和 γ とキャリー Γ を生成し、CSA6 がCSA3 の和をそのキャリーと加算して、和 τ とキャリーT を生成する。これらの値は、折返し条件及び再 *

ID|AG|EX|
mD|AG|EX

...mD|AG|EX (AG中の早期にGate_Cnd_EOPを生成する)

mD|AG|EX (CEOP -> AIR中で有効なNEWID)

ID|AG|EX (EIRは活動状態のNEWIDを有し、それを使用する)

【 0 2 1 7 】例外のためにマイクロコード式レーチンの最中にマイクロコード式例外ハンドラが呼び出された場合、NEWID は、マイクロコード式例外ハンドラの終りに有効にならない。NOT_NEWID_EIR または PIPELINE_HOLD が活動状態の場合は OVERLAP ラッチが保持され、そうでない場合は新しい値がラッチされる。すべてのオーバーラップ BR LO 条件のラッチングは制御記憶機構内で実行される。詳細についてはそのワークブックを参照のこと。オーバーラップ条件の非ラッチ版は論理図に示してある。

* マップ条件での補正に必要なものである。

【 0 2 1 5 】様々な補正のケース用の数式1 4 2 を生成する準備として、CSA4、CSA5、CSA6、CSA7 によってそれぞれ生成された和 ν 、 γ 、 τ 、 σ とキャリーN、 Γ 、T、 ω の様々な組み合わせが、 κ_{56} と共に排他的OR される。これと並行して、 $\sigma (56 : 63)$ と $\omega (57 : 63) || 0$ がキャリー生成器に供給されて ϕ_{56} を生成し、これを数式1 4 3 に従って κ_{56} 及び ω_{56} と組み合わせることによって C_{56} が生成できる。

10 【 数1 4 2 】

$S \pm 0$

【 数1 4 3 】

$$C_{56} = (\omega_{56} \vee \kappa_{56}) \vee \phi_{56}$$

【 0 2 1 6 】この計算は、図5 及び6 を見るとわかるように、一連の2 元排他的OR によって実行される。次いですべての排他的OR の適当な出力が5 7 W-AND に供給されて、表6 に要約した6 つのケースについてのMVC 型オーバーラップO_MVC を生成する。この適当な値は、図5 及び6 にOVERLAP として示す正しいMVC 型オーバーラップ条件を生成するため、図4 に示す復号器によって生成されるCOR (1 : 5) によって選択される。次いでOVERLAP が、後続サイクルで実行すべく2 つのマイクロ命令のうちの1 つを選ぶために制御記憶機構の次のアドレス指定論理機構に提示するため、分岐低マルチプレクサ (図5 及び6 には示さず) に供給される。OVERLAP はまた、その命令サイクル中の後の時間にこの条件が後続のマイクロ命令にとって利用できるようにするためにラッチされる。このラッチは、EIR レジスタから取り出されるNEWID ビットによってリセットされる。このNEWID_EIR は制御記憶論理機構によってセットアップされる。このNEWID 信号のタイミングは、次の通りである。

50 $r a i$ D_1 を格納するRAレジスタの i 番目のビット。

$l h i$ B_2 を格納するLHレジスタの i 番のビット。
 $l a i$ B_1 を格納するLAレジスタの i 番のビット。
 $r h i$ D_2 を格納するRHレジスタの i 番のビット。これは1 2 ビット・レジスタであると仮定し、そのように番号をつけてある。このレジスタを加数として使用する前に左に0 を5 6 個連結すると、AGEN が発生する。

【 0 2 1 8 】OVERLAP の判定は、下記の方程式によって完全に指定される。これらの式では下記の表記法を使用する。

61

これは1 2ビット・レジスタであると仮定し、そのよう * する。
に番号をつけてある。このレジスタを加数として使用する 【 数1 4 4 】
前に左に0 を5 6 個連結すると、AGENが発生す *

$$e_i = lh_i \vee la_i$$

$$e_i = lh_i \vee la_i \vee ra_{(i-52)}$$

$$f_i = lh_i + \overline{la_i} = \overline{lh_i} la_i$$

$$f_i = lh_i \overline{la_i} + lh_i \overline{ra_{(i-52)}} + \overline{la_i} \overline{ra_{(i-52)}}$$

62

$$0 \leq i \leq 51$$

$$52 \leq i \leq 63$$

$$1 \leq i \leq 51$$

$$52 \leq i \leq 63$$

【 数1 4 5 】

10

$$g_i = e_i \vee f_{i+1}$$

$$g_i = e_i \vee f_{i+1} \vee rh_{(i-52)}$$

$$g_{63} = e_{63} \vee rh_{(11)}$$

$$h_i = e_i f_{i+1}$$

$$h_i = e_i f_{i+1} + e_i rh_{(i-52)} + f_{i+1} rh_{(i-52)}$$

$$h_{63} = e_{63} + rh_{(11)}$$

$$0 \leq i \leq 51$$

$$52 \leq i \leq 62$$

$$1 \leq i \leq 51$$

$$52 \leq i \leq 62$$

【 数1 4 6 】

$$\delta_i = g_i$$

$$0 \leq i \leq 39$$

$$\epsilon_{_n i} = e_i + f_{i+1}$$

$$1 \leq i \leq 39$$

【 数1 4 7 】

$$\sigma_i = g_i \vee h_{i+1}$$

$$\sigma_i = g_i \vee h_{i+1} \vee l_{(i-56)}$$

$$\sigma_{63} = g_{63} \vee l_7$$

$$\omega_i = g_i h_{i+1}$$

$$\omega_i = g_i h_{i+1} + g_i l_{(i-56)} + h_{i+1} l_{(i-56)}$$

$$\omega_{63} = g_{63} + l_7$$

$$0 \leq i \leq 55$$

$$56 \leq i \leq 62$$

$$1 \leq i \leq 55$$

$$56 \leq i \leq 62$$

【 数1 4 8 】

$$\tau_i = g_i \vee \epsilon_{_n i+1}$$

$$0 \leq i \leq 38$$

$$T_i = g_i + \epsilon_{_n i+1}$$

$$1 \leq i \leq 38$$

【 数1 4 9 】

40

$$\gamma_i = g_i \vee h_{i+1}$$

$$i = 32, 39$$

$$\Gamma_i = g_i + h_{i+1}$$

$$i = 32, 39$$

【 数1 5 0 】

$$\nu_i = g_i \vee \epsilon_{_n i+1}$$

$$i = 31, 38$$

$$N_i = g_i + \epsilon_{_n i+1}$$

$$i = 31, 38$$

【 数1 5 1 】

$$\begin{aligned}
(\kappa H)_i &= g_i \forall h_{i+1} & 56 \leq i \leq 63 \\
(\kappa G)_{i_N} &= \overline{g_i h_{i+1}} & 56 \leq i \leq 62 \\
(\kappa T)_{i_N} &= \overline{g_i + h_{i+1}} & 56 \leq i \leq 62 \\
(\kappa C)_{63_N} &= \overline{g_{63}} \\
(\kappa G)_{i+1}^{i+1} &= \overline{(\kappa G)_{i_N} ((\kappa T)_{i_N} + (\kappa G)_{i-1_N})} & 56 \leq i \leq 61 \\
(\kappa T)_{i+1}^{i+1} &= \overline{(\kappa T)_{i_N} + (\kappa T)_{i+1_N}} & 56 \leq i \leq 61 \\
(\kappa C)_{62} &= \overline{(\kappa G)_{62_N} ((\kappa T)_{62_N} + (\kappa C)_{63_N})} \\
(\kappa C)_{63} &= \overline{(\kappa C)_{63_N}} \\
(\kappa G)_{i+3}^{i+3} &= \overline{(\kappa G)_{i+1}^{i+1} + (\kappa T)_{i+1}^{i+1} (\kappa G)_{i+2}^{i+2}} & 56 \leq i \leq 59 \\
(\kappa T)_{i+3}^{i+3} &= \overline{(\kappa T)_{i+1}^{i+1} (\kappa T)_{i+2}^{i+2}} & 56 \leq i \leq 59 \\
(\kappa C)_{i_N} &= \overline{(\kappa G)_{i+1}^{i+1} + (\kappa T)_{i+1}^{i+1} (\kappa C)_{i+2}} & i = 60, 61 \\
(\kappa C)_i &= \overline{(\kappa C)_i} & i = 62, 63 \\
\kappa_i &= \overline{(\kappa G)_{i+3}^{i+3} ((\kappa T)_{i+3}^{i+3} + (\kappa C)_{i+4_N})} & 57 \leq i \leq 59 \\
(\kappa C)_i &= \overline{(\kappa C)_{i_N}} & 60 \leq i \leq 63 \\
\kappa_{56} &= \overline{(\kappa G)_{58}^{58} ((\kappa T)_{58}^{58} + (\kappa C)_{60_N})} \\
S_i &= (\kappa C)_{i+1} \forall (\kappa H)_i & 57 \leq i \leq 63 \\
S_{56} &= \kappa_{57} \forall (\kappa H)_{56} \\
S_{63_N} &= \overline{S_{63}}
\end{aligned}$$

【 数1 5 2 】

$$\begin{aligned}
(\phi G)_{i_N} &= \overline{\sigma_i \omega_{i+1}} & 56 \leq i \leq 62 \\
(\phi T)_{i_N} &= \overline{\sigma_i + \omega_{i+1}} & 56 \leq i \leq 62 \\
(\phi C)_{63_N} &= \overline{\sigma_{63} \omega_{64}} \\
(\phi G)_{i+1}^{i+1} &= \overline{(\phi G)_{i_N} ((\phi T)_{i_N} + (\phi G)_{i+1_N})} & i = 56, 58, 60 \\
(\phi T)_{i+1}^{i+1} &= \overline{(\phi T)_{i_N} + (\phi T)_{i+1_N}} \\
(\phi C)_{62} &= \overline{(\phi G)_{62_N} ((\phi T)_{62_N} + (\phi C)_{63_N})} \\
(\phi G)_{58}^{58} &= \overline{(\phi G)_{56}^{56} + (\phi T)_{56}^{56} (\phi G)_{58}^{58}} \\
(\phi T)_{58}^{58} &= \overline{(\phi T)_{56}^{56} (\phi T)_{58}^{58}} \\
(\phi C)_{60_N} &= \overline{(\phi G)_{60}^{60} + (\phi T)_{60}^{60} (\phi C)_{62}} \\
\phi_{56} &= \overline{(\phi G)_{58}^{58} ((\phi T)_{58}^{58} + (\phi C)_{60_N})}
\end{aligned}$$

【 数1 5 3 】

XR 及び XR 8

$$C_{56} = \kappa_{56} \forall \omega_{56} \forall \phi_{56}$$

$$S_{\text{OCOR1}}^{\perp} = (\sigma_0 \forall \omega_1) (\sigma_1 \forall \omega_2) \dots (\sigma_{54} \forall \omega_{55}) (\sigma_{55} \forall \kappa_{56})$$

$$O_{\text{MVCCOR1}} = S_{\text{OCOR1}}^{\perp} C_{56}$$

*【 数1 5 5 】

【 数1 5 4 】

$$\begin{aligned}
S_{\text{OCOR2}}^{\perp} &= (\sigma_0 \forall \omega_1) (\sigma_1 \forall \omega_2) \dots (\sigma_{35} \forall \omega_{36}) (\sigma_{36} \forall \omega_{37}) (\sigma_{37} \forall \omega_{38}) (\gamma_{38} \forall \Gamma_{39}) (\gamma_{39} \forall \omega_{40}) (\sigma_{40} \forall \omega_{41}) (\sigma_{41} \forall \omega_{42}) \dots \\
&\quad (\sigma_{54} \forall \omega_{55}) (\sigma_{55} \forall \kappa_{56})
\end{aligned}$$

$$O_{\text{MVCCOR2}} = S_{\text{OCOR2}}^{\perp} C_{56}$$

【 数1 5 6 】

$$S_{\text{OCOR3}}^{\perp} = (\tau_0 \forall T_1) (\tau_1 \forall T_2) \dots (\tau_{35} \forall T_{37}) (\tau_{37} \forall T_{38}) (\tau_{38} \forall \Gamma_{39}) (\gamma_{39} \forall \omega_{40}) (\sigma_{40} \forall \omega_{41}) (\sigma_{41} \forall \omega_{42}) \dots (\sigma_{54} \forall \omega_{55}) (\sigma_{55} \forall \kappa_{56})$$

$$O_{\text{MVCCOR3}} = S_{\text{OCOR3}}^{\perp} C_{56}$$

【 数1 5 7 】

$$S_{0COR4}^1 = (\sigma_0 \vee \omega_1) (\sigma_1 \vee \omega_2) \dots (\sigma_{28} \vee \omega_{29}) (\sigma_{29} \vee \omega_{30}) (\sigma_{30} \vee N_{31}) (\gamma_{31} \vee \Gamma_{32}) (\gamma_{32} \vee \omega_{33}) (\sigma_{33} \vee \omega_{34}) (\sigma_{34} \vee \omega_{35}) \dots (\sigma_{34} \vee \omega_{35}) (\sigma_{35} \vee \kappa_{36})$$

$$O_{MVCCOR4} = S_{0COR4}^1 C_{56}$$

【 数1 5 8 】

$$S_{0COR5}^1 = (\tau_0 \vee T_1) (\tau_1 \vee T_2) \dots (\tau_{28} \vee T_{29}) (\tau_{29} \vee T_{30}) (\tau_{30} \vee \Gamma_{31}) (\gamma_{31} \vee \omega_{32}) (\sigma_{32} \vee \omega_{33}) (\sigma_{33} \vee \omega_{34}) (\sigma_{34} \vee \omega_{35}) \dots (\sigma_{34} \vee \omega_{35}) (\sigma_{35} \vee \kappa_{36})$$

$$O_{MVCCOR5} = S_{0COR5}^1 C_{56}$$

【 数1 5 9 】

SAR1HI 12ビット加算検出

$$SAR1H_i = \overline{1a_i \vee ra_{ij-52}} \quad 52 \leq i \leq 55$$

$$SAR1G_{i_N} = \overline{1a_i \vee ra_{ij-52}} \quad 52 \leq i \leq 62$$

$$SAR1T_{i_N} = \overline{1a_i \vee ra_{ij-52}} \quad 52 \leq i \leq 62$$

$$SAR1C_{53_N1} = \overline{1a_{53} \vee ra_{11}}$$

$$SAR1H_{i_N} = \overline{SAR1H_i} \quad 52 \leq i \leq 55$$

$$(SAR1G)_{i+1}^+ = \overline{(SAR1G)_{i_N} ((SAR1T)_{i_N} + (SAR1G)_{i+1_N})} \quad 52 \leq i \leq 61$$

$$(SAR1T)_{i+1}^+ = \overline{((SAR1T)_{i_N} + (SAR1T)_{i+1_N})} \quad 52 \leq i \leq 61$$

$$(SAR1C)_{52} = \overline{(SAR1G)_{52_N} ((SAR1T)_{52_N} + (SAR1C)_{53_N1})}$$

$$(SAR1C)_{53} = \overline{(SAR1C)_{53_N1}}$$

$$(SAR1G)_{i+3_N} = \overline{(SAR1G)_{i+1}^+ + (SAR1T)_{i+1}^+ (SAR1G)_{i+1}^+} \quad 52 \leq i \leq 59$$

$$(SAR1T)_{i+3_N} = \overline{(SAR1T)_{i+1}^+ (SAR1T)_{i+1}^+} \quad 52 \leq i \leq 59$$

$$(SAR1C)_{i_N2} = \overline{(SAR1G)_{i+1}^+ + (SAR1T)_{i+1}^+ (SAR1C)_{i+1}^+} \quad i = 60, 61$$

$$(SAR1C)_{i_N2} = \overline{(SAR1C)_{i+1}^+} \quad i = 62, 63$$

$$(SAR1G)_{i+7}^+ = \overline{(SAR1G)_{i+3_N} ((SAR1T)_{i+3_N} + (SAR1G)_{i+3_N})} \quad 52 \leq i \leq 55$$

$$(SAR1T)_{i+7}^+ = \overline{(SAR1T)_{i+3_N} + (SAR1T)_{i+3_N}} \quad 52 \leq i \leq 55$$

$$(SAR1C)_{56} = \overline{(SAR1G)_{56_N} ((SAR1T)_{56_N} + (SAR1C)_{60_N2})} \quad 56$$

$$(SAR1C)_{i+} = \overline{(SAR1C)_{i_N2}} \quad 60 \leq i \leq 63$$

$$(SAR1C)_{i_N} = \overline{(SAR1G)_{i+7}^+ + (SAR1T)_{i+7}^+ (SAR1C)_{i+7}^+} \quad 52 \leq i \leq 55$$

$$(SAR1C)_{56_N} = \overline{(SAR1C)_{56}}$$

$$SAR1_{i_N} = (SAR1C)_{i+1_N} \vee SAR1H_i \quad 52 \leq i \leq 55$$

$$SAR1_i = (SAR1C)_{i+1_N} \vee SAR1H_{i_N} \quad 52 \leq i \leq 55$$

$$SAR1_{52_to_55_ZEROS} = (SAR1_{52_N}) (SAR1_{53_N}) (SAR1_{54_N}) (SAR1_{55_N})$$

$$SAR1_{52_to_55_ONES} = (SAR1_{52}) (SAR1_{53}) (SAR1_{54}) (SAR1_{55})$$

$$(SAR1C)_{52} = \overline{(SAR1C)_{52_N}}$$

【 数1 6 0 】

SAR1HI、SAR1LO、WRAP1 検出論理

$$la_i_N = \overline{la_i}$$

$$33 \leq i \leq 51$$

$$la_40_TO_51_ZEROS = la_{40_N} la_{41_N} \dots la_{50_N} la_{51_N}$$

$$la_40_TO_50_ONES = la_{40} la_{41} \dots la_{50}$$

$$la_40_TO_51_ONES = la_40_TO_50_ONES la_{51}$$

$$la_40_TO_50_ONES_A_51_ZERO = la_40_TO_50_ONES la_{51_N}$$

$$la_33_TO_39_ZEROS = la_{33_N} la_{34_N} \dots la_{39_N} la_{39_N}$$

$$la_33_TO_39_ONES = la_{33} la_{34} \dots la_{38} la_{39}$$

$$la_33_TO_50_ONES_A_51_ZERO = la_33_TO_39_ONES la_40_TO_50_ONES_A_51_ZERO$$

$$la_33_TO_51_ZEROS = la_33_TO_39_ZEROS la_40_TO_51_ZEROS$$

$$la_33_TO_51_ONES = la_33_TO_39_ONES la_40_TO_51_ONES$$

$$SAR1_TO_51_0S = 24_BIT_MD la_40_TO_51_ZEROS + 31_BIT_MD la_33_TO_51_ZEROS$$

$$SAR1_TO_51_1S = 24_BIT_MD la_40_TO_51_ONES + 31_BIT_MD la_33_TO_51_ONES$$

$$SAR1_TO_50_1S_A_51_0 = 24_BIT_MD la_40_TO_50_ONES_A_51_ZERO + \\ 31_BIT_MD la_33_TO_50_ONES_A_51_ZERO$$

$$SAR1LO = [(SAR1C)_{32_N} SAR1_TO_51_0S + (SAR1C)_{32} SAR1_TO_51_1S] SAR1_52_TO_55_ZEROS$$

$$SAR1HI = [(SAR1C)_{32_N} SAR1_TO_51_1S + (SAR1C)_{32} SAR1_TO_50_1S_A_51_0] SAR1_52_TO_55_ONES$$

$$WRAP1 = (SAR1C)_{32} SAR1_TO_51_1S$$

$$WRAP1_N = \overline{WRAP1}$$

【 数1 6 1 】

SAR2HI 12ビット加算検出

$$\begin{aligned}
SAR2H_i &= lh_i \vee rh_{i-52} & 52 \leq i \leq 55 \\
SAR2G_{i_N} &= \overline{lh_i rh_{i-52}} & 52 \leq i \leq 62 \\
SAR2T_{i_N} &= \overline{lh_i + rh_{i-52}} & 52 \leq i \leq 62 \\
SAR2C_{63_N1} &= \overline{lh_{63} rh_{11}} \\
SAR2H_{i_N} &= \overline{SAR2H_i} & 52 \leq i \leq 55 \\
(SAR2G)_{i+1} &= \overline{(SAR2G)_{i_N} ((SAR2T)_{i_N} + (SAR2G)_{i+1_N})} & 52 \leq i \leq 61 \\
(SAR2T)_{i+1} &= \overline{((SAR2T)_{i_N} + (SAR2T)_{i+1_N})} & 52 \leq i \leq 61 \\
(SAR2C1)_{62} &= \overline{(SAR2G)_{62_N} ((SAR2T)_{62_N} + (SAR2C)_{63_N1})} \\
(SAR2C1)_{63} &= \overline{(SAR2C)_{63_N1}} \\
(SAR2G)_{i+1_N} &= \overline{(SAR2G)_{i+1} + (SAR2T)_{i+1} (SAR2G)_{i+1}} & 52 \leq i \leq 59 \\
(SAR2T)_{i+1_N} &= \overline{(SAR2T)_{i+1} (SAR2T)_{i+1}} & 52 \leq i \leq 59 \\
(SAR2C)_{i_N2} &= \overline{(SAR2G)_{i+1} + (SAR2T)_{i+1} (SAR2C1)_{i+2}} & i = 60, 61 \\
(SAR2C)_{i_N2} &= \overline{(SAR2C1)_i} & i = 62, 63 \\
(SAR2G)_{i+1} &= \overline{(SAR2G)_{i+1_N} ((SAR2T)_{i+1_N} + (SAR2G)_{i+1_N})} & 52 \leq i \leq 55 \\
(SAR2T)_{i+1} &= \overline{(SAR2T)_{i+1_N} + (SAR2T)_{i+1_N}} & 52 \leq i \leq 55 \\
(SAR2C)_{56} &= \overline{(SAR2G)_{56_N} ((SAR2T)_{56_N} + (SAR2C)_{60_N2})} & 56 \\
(SAR2C)_i &= \overline{(SAR2C)_{i_N2}} & 60 \leq i \leq 63 \\
(SAR2C)_{i_N} &= \overline{(SAR2G)_{i+1} + (SAR2T)_{i+1} (SAR2C)_{i+2}} & 52 \leq i \leq 55 \\
(SAR2C)_{56_N} &= \overline{(SAR2C)_{56}} \\
SAR2_{i_N} &= (SAR2C)_{i+1_N} \vee SAR2H_i & 52 \leq i \leq 55 \\
SAR2_i &= (SAR2C)_{i+1_N} \vee SAR2H_{i_N} & 52 \leq i \leq 55 \\
SAR2_{52_to_55_ZEROS} &= (SAR2_{52_N}) (SAR2_{53_N}) (SAR2_{54_N}) (SAR2_{55_N}) \\
SAR2_{52_to_55_ONES} &= (SAR2_{52}) (SAR2_{53}) (SAR2_{54}) (SAR2_{55}) \\
(SAR2C)_{52} &= \overline{(SAR2C)_{52_N}}
\end{aligned}$$

SAR2HI、SAR2LO、WRAP2 検出論理

$$\begin{aligned}
 lh_i_N &= \overline{lh_i} & 33 \leq i \leq 51 \\
 lh_40_TO_51_ZEROS &= lh_{40_N} lh_{41_N} \dots lh_{50_N} lh_{51_N} \\
 lh_40_TO_50_ONES &= lh_{40} lh_{41} \dots lh_{50} \\
 lh_40_TO_51_ONES &= lh_40_TO_50_ONES lh_{51} \\
 lh_40_TO_50_ONES_A_51_ZERO &= lh_40_TO_50_ONES lh_{51_N} \\
 lh_33_TO_39_ZEROS &= lh_{33_N} lh_{34_N} \dots lh_{39_N} lh_{39_N} \\
 lh_33_TO_39_ONES &= lh_{33} lh_{34} \dots lh_{39} lh_{33} \\
 lh_33_TO_50_ONES_A_51_ZERO &= lh_33_TO_39_ONES lh_40_TO_50_ONES_A_51_ZERO \\
 lh_33_TO_51_ZEROS &= lh_33_TO_39_ZEROS lh_40_TO_51_ZEROS \\
 lh_33_TO_51_ONES &= lh_33_TO_39_ONES lh_40_TO_51_ONES \\
 SAR2_TO_51_0S &= 24_BIT_MD lh_40_TO_51_ZEROS + 31_BIT_MD lh_33_TO_51_ZEROS \\
 SAR2_TO_51_1S &= 24_BIT_MD lh_40_TO_51_ONES + 31_BIT_MD lh_33_TO_51_ONES \\
 SAR2_TO_50_1S_A_51_0 &= 24_BIT_MD lh_40_TO_50_ONES_A_51_ZERO + \\
 & \quad 31_BIT_MD lh_33_TO_50_ONES_A_51_ZERO \\
 SAR2LO &= [(SAR2C)_{52_N} SAR2_TO_51_0S + (SAR2C)_{52} SAR2_TO_51_1S] SAR2_52_TO_55_ZEROS \\
 SAR2HI &= [(SAR2C)_{52_N} SAR2_TO_51_1S + (SAR2C)_{52} SAR2_TO_50_1S_A_51_0] SAR2_52_TO_55_ONES \\
 WRAP2 &= (SAR2C)_{52} SAR2_TO_51_1S \\
 WRAP2_N &= \overline{WRAP2}
 \end{aligned}$$

【 数1 6 3 】

COR (1 : 5) 生成論理

$$\begin{aligned}
 REMAP1_N &= \overline{SAR2HI SAR1LO} \\
 REMAP2_N &= \overline{SAR2LO SAR1HI} \\
 REMAP1 &= \overline{REMAP1_N} \\
 REMAP2 &= \overline{REMAP2_N} \\
 COR(1) &= 64_BIT_MD + REMAP1_N WRAP1_N WRAP2_N + WRAP2 WRAP1 \\
 COR(2) &= 24_BIT_MD REMAP1_N WRAP2_N WRAP1 \\
 COR(3) &= 24_BIT_MD WRAP2 WRAP1_N + 24_BIT_MD REMAP1 WRAP1_N \\
 COR(4) &= 31_BIT_MD REMAP1_N WRAP2_N WRAP1 \\
 COR(5) &= 31_BIT_MD WRAP2 WRAP1_N + 31_BIT_MD REMAP1 WRAP1_N
 \end{aligned}$$

【 数1 6 4 】

OVERLAP 生成論理

$$\begin{aligned}
 OVERLAP &= COR(1) O_{MYCCOR1} + COR(2) O_{MYCCOR2} + COR(3) O_{MYCCOR3} + COR(4) O_{MYCCOR4} + \\
 & \quad COR(5) O_{MYCCOR5}
 \end{aligned}$$

【 図面の簡単な説明】

【 図1 】 記憶オペランドが破壊的記憶オペランド・オーバーラップを有する可能性のある、すべての独自ケースを示す図である。

【 図2 】 記憶オペランドが破壊的記憶オペランド・オー

バーラップを有しない、すべての独自ケースを示す図である。

【 図3 】 記憶オペランドが破壊的記憶オペランド・オーバーラップを有しない、すべての独自ケースを示す図である。

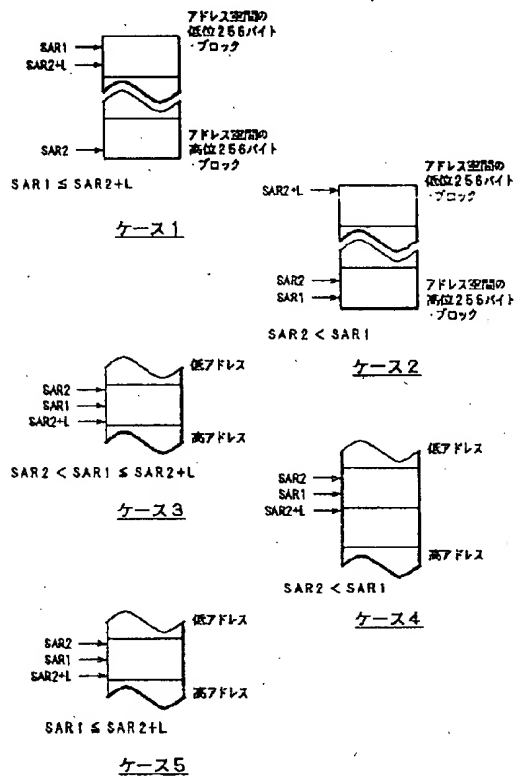
【図4】破壊的記憶オペランド・オーバーラップを予測するのに必要なハードウェアの概要を示す図である。

【図5】図3の概要に含まれるオーバーラップ予測論理ブロックの詳細を示す図である。

【図6】図3の概要に含まれるオーバーラップ予測論理ブロックの詳細を示す図である。

【図7】宛先記憶オペランド・アドレスが仮想アドレス空間の最高位256バイト・ブロックに常駐し、仮想アドレス空間の最低位256バイト・ブロックに常駐する

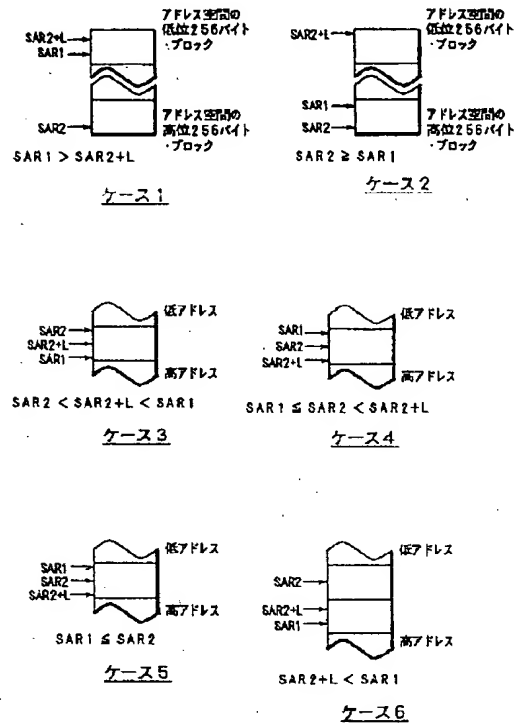
【図1】



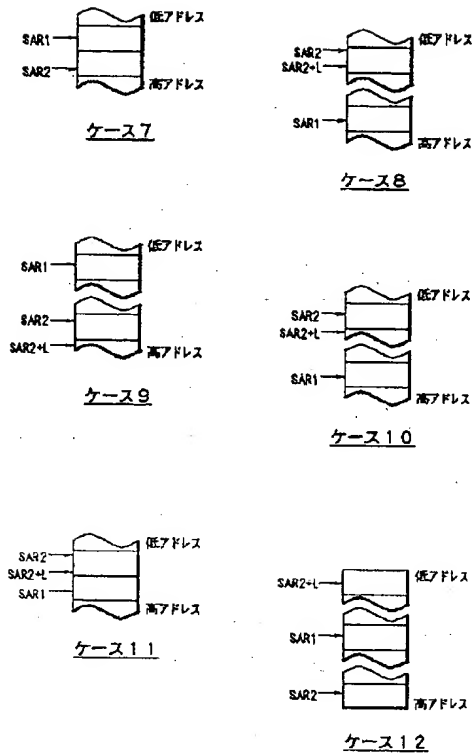
かどうか、および宛先記憶オペランド・アドレスの計算中に折返しが発生する条件を決定するための論理の詳細を示す図である。

【図8】ソース記憶オペランド・アドレスが仮想アドレス空間の最高位256バイト・ブロックに常駐し、仮想アドレス空間の最低位256バイト・ブロックに常駐するかどうか、およびソース記憶オペランド・アドレスの計算中に折返しが発生する条件を決定するための論理の詳細を示す図である。

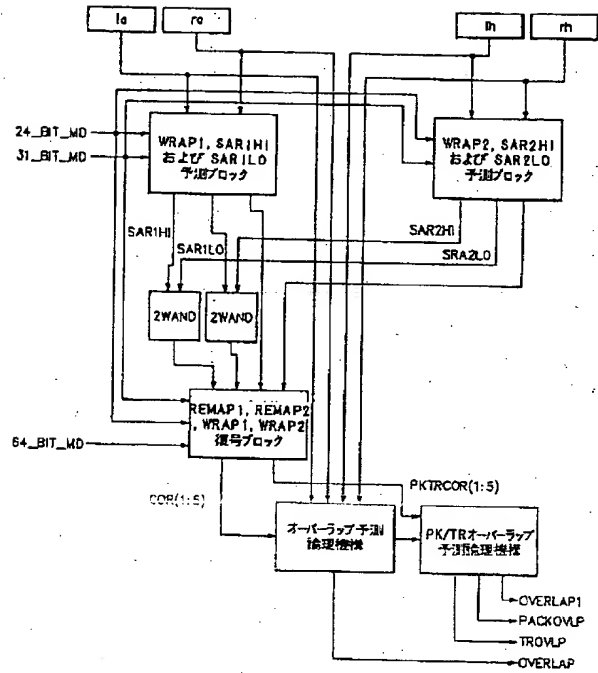
【図2】



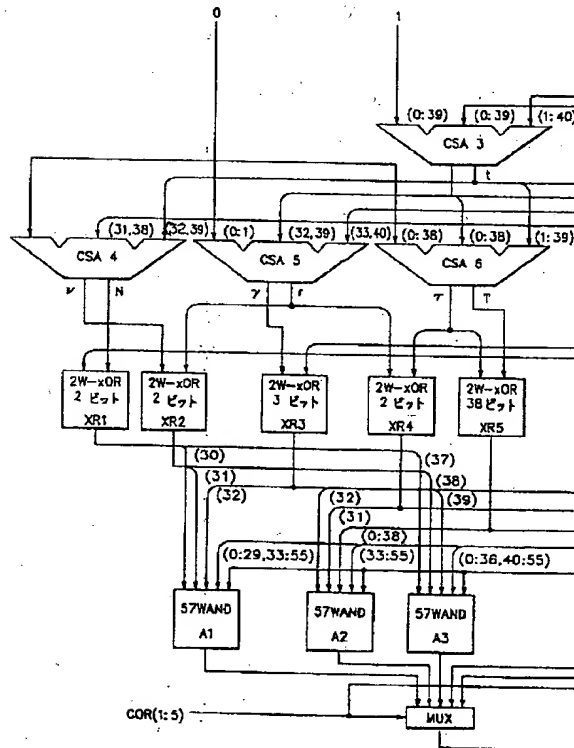
【 図3 】



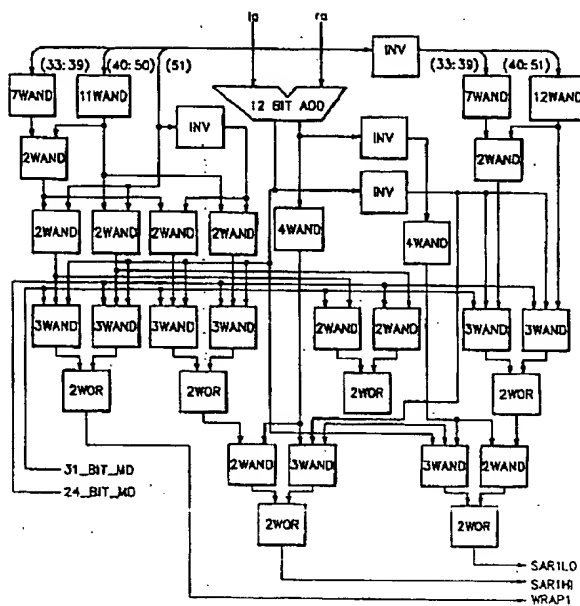
【 図4 】



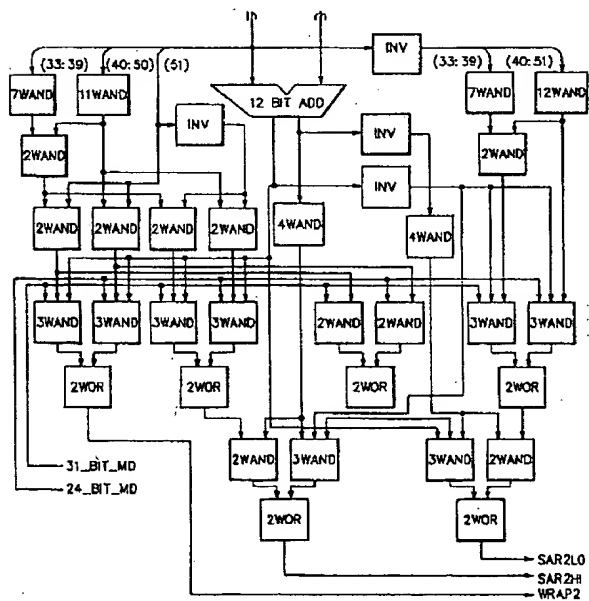
【 図5 】



【 図7 】



【 図8 】



フロント ページの続き

(72)発明者 スタマティス・ヴァッシリアディス
アメリカ合衆国13850、ニューヨーク州ヴ
ェスタル、ヴェスタル・ロード 717